整体数字化建设项目

通用权限管理系统组件
使用说明书



开发单位：杭州海日涵科技

开发部门：产品部

编制日期：2011年03月



使用说明书

模块名称：通用权限管理模块	**模块编号：**1
项目负责人：吉日嘎拉	**所属部门：**产品部
文档编制：陈波涛、陆云钢	**编制日期：**2009-01-14
文档审核：袁华林	**审核日期：**2010-03-01
文档批准：吉日嘎拉	**批准日期：**2011-03-11
版 本 号：3.3	**修 正 号：**04



修改记录

日期	版本	修改说明	修改者
2008-01-29	0.1	创建文档。	陈波涛
2008-04-28	1.0	完善文字描述。	陆云钢
2008-05-14	1.1	完善图标格式等。	陆云钢
2009-11-11	2.0	更新功能界面图片，添加图片名称，调整文档顺序，编写新增功能页面文档（申请帐号、只显示有效、基础编码管理和修改密码）。	狄永
2010-04-01	3.0	安装方法等进行补充完善。	吉日嘎拉
2010-05-08	3.0	重新抓图、文字校对	吉日嘎拉
2010-06-03	3.1	修改文档中 35 处错误	梁帮友
2010-10-22	3.2	重新排版、文字校对	段飞
2011-03-11	3.3	更新说明书抓图更新	吉日嘎拉



目录





整体数字化建设项目通用权限管理组件

第1章 通用权限管理组件简介

本软件模块的严谨设计定位、精心编码实现、不断维护推广、持续优化改进，主要是为了实现一个可以高度重复利用劳动成果的工具软件，并有偿提供给所需的人们，另想成为国人值得骄傲的知名软件功能模块，可供国内管理类开发人员在日常工作中进行灵活二次开发利用的模块，开发管理类软件的必备工具之一，同时本软件的系统架构可供广大业余兴趣爱好者模仿。

本系统模块能提高管理类软件系统的整体质量、提高模块与模块之间的兼容性、提高代码的重复利用率，能够使软件系统架构更加合理、质量更加过硬。从而使得劳动成果最大程度上重复利用。

有错就修改，每天努力一点点，提高一点点，坚持不懈，不断努力每一天！

1.1 软件开发现状分析

新员工没人带，老员工忙着救项目、就算是有空老员工也有疲倦的时候，也有不愿意带人的，也有不愿意把 简单的东西反反复复教人的，一年又一年，一批又一批人换来换去，很有可能也懒的带。

有些人喜欢与别人交流，但是大部分 IT 软件开发人员习惯了与计算机沟通，很可能有些不适应与人沟通交流，或者忙着解决他在技术上遇到的难题等。

1.2 功能定位、建设目标

一） 实现简易的单点登录功能，用户只要记住一对用户名密码就可以：

随着信息化项目、数字化项目越来越多的被研发实施，需要记住的用户名密码变得越来越多，随着公司业务规模的扩展、公司的发展壮大或者本身是一个大型的集团公司，内部前后实施的信息化系统会越来越多，这时也需要记忆越来越多的用户名密码，

而且后台的管理越来越变得复杂，甚至是变成累赘。

二） 多个管理系统可以用统一的一套后台管理工具进行管理：

管理人员、开发人员、测试人员、实施人员，反复要开发 N 多个大同小异的后台管理工具、开发、测试、修改错误、改进功能，没完没了，全国各地 N 多的公司，N 多的开发人员，在这个没多大价值的事情上浪费了 N 多的精力，实施维护人员还要安装、实施、学习、维护 N 多个大同小异的后台管理工具，需要很高的学习与维护成本

三） 建立信息化数字化项目的数据中心，提高基础数据的重复利用率：

随着同一个公司内部实施的信息化系统数量的增加、遇到了将同样的基础数据录入到 N 多个管理系统中的问题、设想一下，将一个用户信息录入到 10-20 个系统里，有一个基础数据有变化，需要同时在 N 多个相应的系统里进行修正，这是一个很复杂且毫无意义的、毫无创新的重复工作。

四） 提高同一个公司内开发的多个软件项目的兼容性、可维护性：

有规模的软件公司内部大多是采用项目组的形式管理和开展日常开发工作；但是由于沟通能力、沟通技巧、沟通效率、项目时间紧等各种复杂的原因，最终会导致各个项目组之间的分工、沟通、交流不充分，各个项目组之间开发出来的管理系统都不能互相兼容，功能模块都存在重复建设、分工不明确、职责不明确等问题。

五） 努力实现各软件公司研发的功能模块的可集成性：

同一个大型客户，往往前后会购买多个软件公司生产的功能模块、信息管理系统，但是这些公司之间的功能模块往往也会存在不兼容的问题，数据重复录入，基础信息无法共享的问题，管理维护多个软件提供商提供的产品、功能模块，也带来了繁杂的后期维护、功能集成方面不兼容的后患。

六） 成为广大软件开发爱好者开发管理类软件的基础组件之一，提高生产力：

专业软件公司的信息化系统，往往是多个人维护了多年，不仅仅是功能复杂、关联繁杂，这时若遇到开发一个小型的软件项目或者一个简单的软件项目，往往没有能力在短时间内干净利落的弄好，就像俗话讲的麻雀虽小五脏俱全。

本系统提供一个与任何业务系统毫无关系的、功能定位明确的轻量级的快速开发平台。

本系统有严谨的数据库设计文档、有比较完善的配套参考资料、功能讲解，同时有偿提供全部的源码、注释及配套工具及说明文档。

七） 成为管理类软件开发爱好者的入门者学习提高的标准教程：

本系统是符合严格的分层原则、设计模式、数据库设计规范、编码精良，思路严谨的解决多种数据库兼容性问题的小型软件项目。

同时本系统又是日后开发管理系统的基础的必要组成部分，将来可以派上用处的可能性很大，值得学习，值得长期维护。

编码简单易懂，方便初学者学习入门。

编码简单易懂，方便后期的维护改进。

1.3 目标用户群体

一）政府机构、集团公司：

可成为大型软件项目中的统一的组织机构、员工、客户、岗位角色分配、工作流程定义、权限分配的集中规划、配置、日常管理工具；可成为整个组织内的统一的信息化建设的基础数据的建设、管理、维护的工具；基础数据中心的管理维护工具，同时可以成为建设简单高效的单点登录系统的核心组成部分。

可成为众多信息化技术服务提供商的新开发软件项目的规范要求、整体约束的依据及配套工具、应用系统开发的标准参考模型；达到整个组织内部使用的软件产品、软件项目有一定的标准规范、能满足基础信息的共享，管理工具统一化的要求，杜绝相同数据的重复录入、重复管理，提高工作效率。

二）中小型软件开发公司、技术咨询公司：

可以成为统一的、稳定的、标准的、功能完善的后台配置权限管理工具，降低多个软件项目及多个项目组的开发成本、维护成本、项目实施成本，项目管理成本，可以显著提高工作效率，节省开发周期及开发费用投入，减少相应的测试维护改进功能的时间成本，同时可以减少由于开发人员变动带来的项目风险。

同时也可以成为新手的基础参考模型、分层理念、多数据库支持的参考模型、编码规范要求、异常处理、C\S 和 B\S 系统的兼容性设计的参考。

最大程度避免重复开发，原地踏步，反复原地踏步，杜绝每个开发人员和每个开发项目组，都搞一套不兼容的权限管理体系；不兼容的基础用户数据等等，使得公司内部的产品、项目、功能模块之间有更高的兼容性。

三）培训教程、大学课外教程、新员工入职培训材料：

本项目设计严谨、简单易读、编码规范、功能实用，可以作为 IT 培训机构的基础培训内容，同时本项目有一定的通用性及规范性案例的价值，可以作为新员工入职前的基础培训内容之一、编码规范的参考、数据库设计、分层设计、单一职责原则、设计模式等等的入门参考级教程。

能达到节省培训成本，节约在培训方面的投入，有一个统一规范的入职前培训教材，可以减少由于新员工的培训导致对开发中的软件项目的进度、进展的负面影响等。

四）管理类软件开发爱好者：

本项目设计思路严谨、简单易读、编码规范、功能实用，并且是开发管理类软件的基础性的管理工具，有一定的通用性及规范性方面的参考价值，同时可以作为快速开发中小型软件项目的基础，可以通过模仿、复用等显著缩短项目开发周期，达到快速见效的目的。

此工具经过多年的不断改进维护，经过了多个软件项目的实战考验，能满足不同规模软件项目的灵活配置、灵活管理、快速开发、快速实施的要求，是个不错的应急管理工具；同时有成熟完善的组织机构管理、用户管理、角色管理、模块（菜单）管理、权限配置分配管理的设计严谨、运行稳定的功能模块。

1.4 系统优点

一）经过多年的实践不断优化、精心维护、运行稳定、功能完善：

能经得起不同实施策略下客户的折腾，能满足各种情况下客户的复杂需求。

二）编码实现简单易懂、符合设计模式等理念：

上手快，见效快、方便维护，能控制软件项目后期维护风险。

三）有严谨的分层理念、完全符合主流的 **SOA** 理念架构：

程序可以采用不同的实施策略、架构需求，方便维护、方便扩展。

四）符合 **RBAC** 的权限理念、同时也能满足国内小型软件项目的灵活性需求：

不仅符合国际通用标准，又能满足国内的小型软件项目的灵活设置需求。

五）不仅支持多数据库也支持多种开发语言、多语言界面：

支持 B/S、C/S 系统，可以支持 Java 语言开发项目的接口功能，SQLServer、Oracle 数据库等。

六）提供全部的设计文档、源码实现、配套说明文档：

相关设计文档、配套资料可以在不同软件项目中反复使用，提高工作效率。

七）完全支持最终用户的个性化 **2** 次开发实际需求：

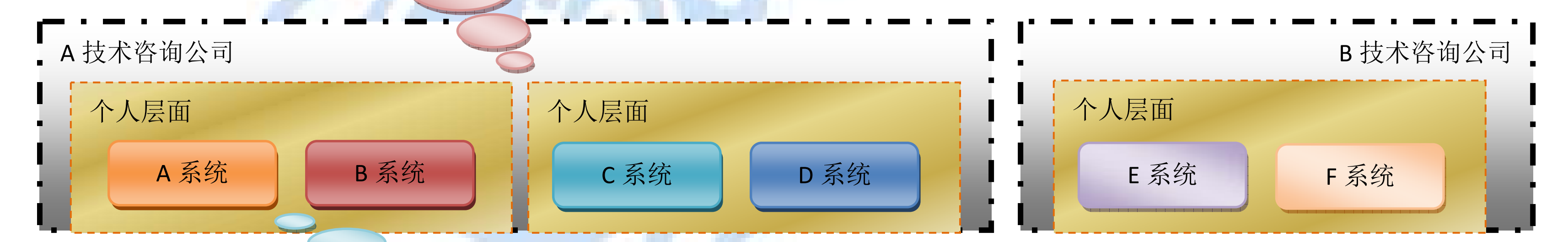
提供完善的接口函数调用说明、开放接口、开放源码、开放数据库结构设计。

1.5 系统架构大局观

3.某个最终用户的各个系统之间的兼容性问题？可维护性问题？数据的共享及可重复利用问题？软件系统之间的协同作业问题？集中管理问题？

2.某个公司开发的多个系统之间的兼容性问题？可维护、可扩展性问题？代码组件的可重复利用问题？版本控制问题？人员变动问题？





1.某个开发人员前前后后若干年开发的各个系统之间的兼容性问题？可维护性问题？代码组件的可重复利用问题？历史数据的重复利用问题？

1.6 物理结构图

权限配置
管理工具

B 核心业务
客户端

权限配置
管理组件

A 分支业务系统

权限接口
二次开发 API

B 核心业务
应用服务

用户中心
【权限数据库】

B 核心业务数据库

1.7 逻辑结构图

数据存储层

业务逻辑层

用户交互层

A 分支业务系统

A 分支业务数据库

A 分支业务应用服务

A 分支业务客户端

用户中心

用户中心应用服务

权限管理配置工具

底层数据模型

C/S 配置管理工具

B 核心业务数据库

B 核心业务应用服务

B 核心业务客户端

标准服务接口、服务实现、多种服务发布方式

1.8 系统运行环境配置

一）管理端：

本系统以微软的 C#.NET 编程语言开发设计，所以需要在微软的网站下下载最新的.NET Framework 运行环境，管理程序需要运行在 Windows 操作系统上。

下载微软的.NET Framework 运行环境请连接下面的网址

http://msdn.microsoft.com/zh-cn/netframework/aa569263.aspx

二）后台数据库：

本系统支持多种数据库系统，例如 SQLServer、Oracle、MySql，可以按用户的使用习惯，任意选择其中一种数据库系统。

目前提供 SQLServer 版本的数据库，其他版本的数据库需要用户自己通过导入导出脚本语言的方式进行处理。

1.9 安装程序、数据库设计、数据库文件

请添加技术支持 QQ：2520 – 56973，索取相应的程序源码、设计文档等等。



1.10 相关程序文档结构说明



图 1-10-1 文档结构图

01. DotNet.Utilities：通用类库部分，经常用的类都封装在这里，不断完善，不断积累，非常好用。

02. DotNet.DbUtilities：数据库访问部分，这里能实现多种数据库的访问，而且实现了换数据库彻底不改代码的功能。

03. DotNet.Model：模型定义部分，主要是系统都处理哪些模型，说俗点儿就是哪些类。

04. DotNet.Business：商业逻辑部分，这里主要是编写核心的商业逻辑，这个积累是很重要的。

05. DotNet.IService：服务接口定义部分，这里主要声明，有哪些服务方法，都提供什么接口。

06. DotNet.Service：服务实现部分：这里就是 SOA 体系的服务程序部分，对外提供的服务，都通过调用这里实现。

07. DotNet.RemotingServer：远程服务部分：主要是实现了 Remoting 的服务器端部分。



08. DotNet.WindowsService：Windows 服务部分：主要是以 Windows 的服务的方式实现具体服务。

09. DotNet.WebService：Web 服务部分：把服务程序以 Web 服务的形式发布部分。

10. DotNet.WebService.Client: Web 服务的客户端调用部分:主要是实现 WebService 的调用实现部分。

11. DotNet.WinForm.Utilities：传统的 C\S 项目部分，通用组件，采用这些组件快速提高开发效率。

12. DotNet.WinForms：传统的 C\S 项目部分，每个子程序可以单独运行，也可以变成母程序的模块。

13. DotNet.WinForm：传统的 C\S 项目部分的主程序部分。

14. DotNet.Web：传统的 B\S 项目部分。

15. DotNet.Example：标准例子程序部分：方便别人学习系统架构，可以快速入门，有简短的样例代码。

16. DotNet.WCFHost： WCF 服务器端用例程序。

17. DotNet.WCFClient：WCF 客户端调用用例程序。



1.11 软件安装、配置、运行方法

一）软件安装：

由于本系统都是针对软件技术开发者，所以没有做成安装包的方式发行，提供源码 + 二次开发调用的 API，动态连接库等方式，只要把数据库附加上去，修改好配置文件即可。

二）配置：

源码文件的文档目录结构如下图，主项目为 DotNet.WinForm， 请在解决方案中将 DotNet.WinForm 项目设置为启动项目。

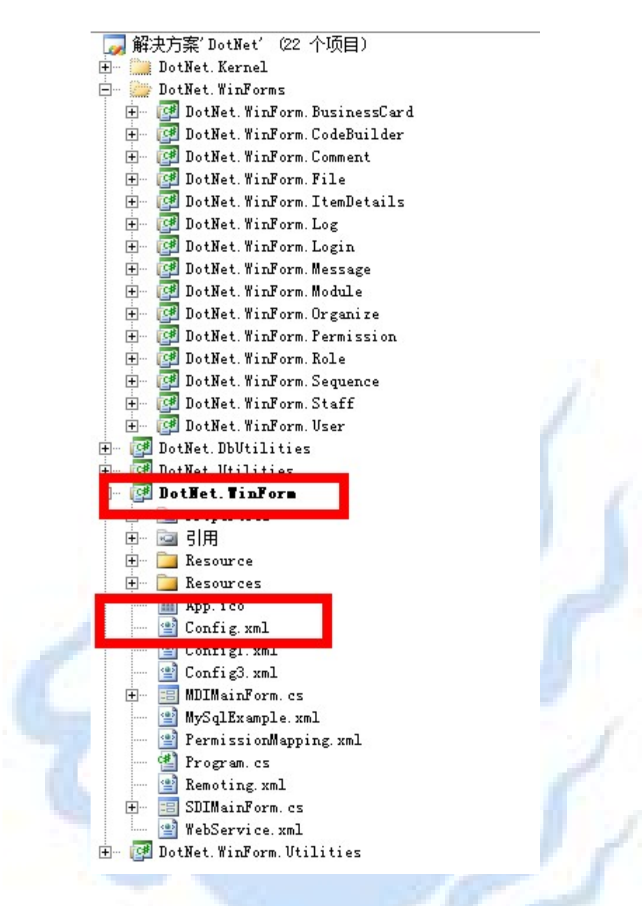


图 1-11-1 配置文件

配置文件的为 Config.xml，设置配置项为 UserCenterDbConnection，单机版参考配置为如下。

```
<add key="UserCenterDbConnection" value="Data Source=JIRIGALA-PC\SQLEXPRESS;Initial Catalog=UserCenter;Integrated Security=SSPI;" />
```

配置文件中附有其他类型数据库的连接串的参考配置。

三）运行方法：

在 Microsoft Visual Studio 开发环境中编译此解决方案源码，编译后的结果如下图

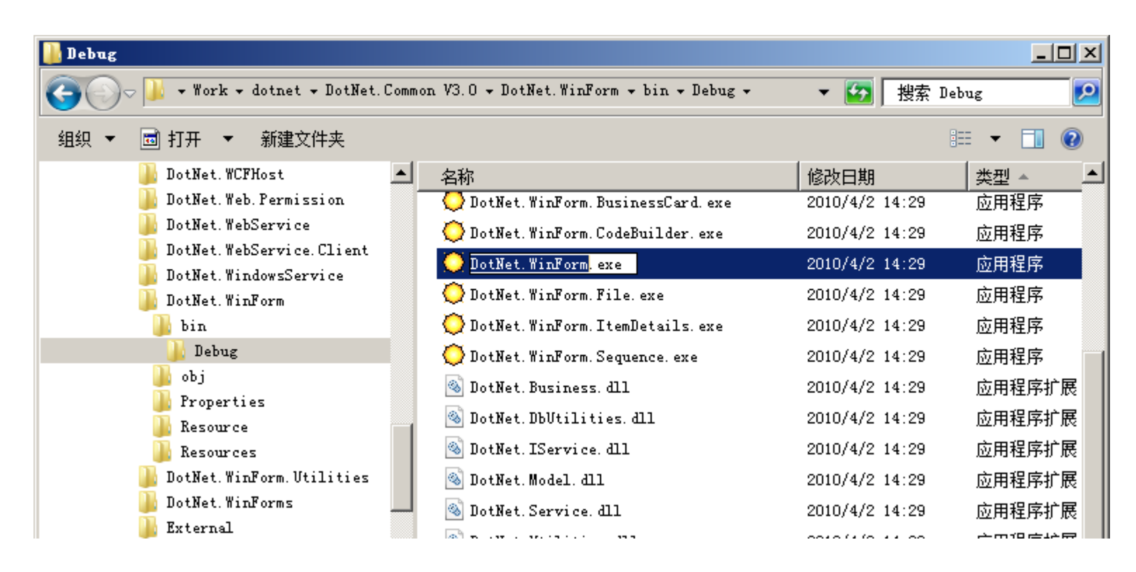


图 1-11-2 主程序

双击可执行主程序 DotNet.WinForm.exe；若数据库连接配置正确，会出现如下登录界面，如下图。

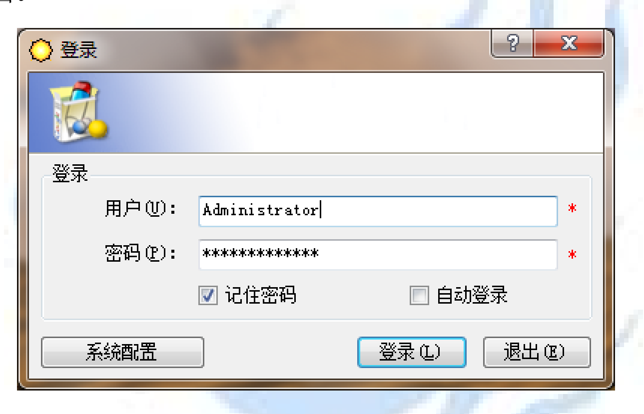


图 1-11-3 登录界面

默认情况下，超级管理员的密码为空，点击登录进入后台管理软件。正确登录后会在操作系统的任务栏右下角区域会出现小太阳的图标，如图 1-2 其中黄色小圆图标即为后台管理程序。

鼠标右键单击黄色小图标，出现软件主菜单，根据需求选择自己需要的功能，如用户（帐

号）管理功能等等。如图下图。



图 1-11-4 右键系统菜单

第2章 登录后台管理系统

很早的时候曾听别人说，印度软件公司中有些软件开发人员可能一直负责开发系统登录功能，当时觉得不就是一个登录系统的功能嘛？把用户名、密码输入好后按一下登录按钮不就可以了？哪里有什么那么多好做的功能？后来随着对管理软件的深入理解、接触过很多客户后才真正体会到，真正做好一个登录窗体也的确不是那么容易的事情，一个软件的质量是否好？设计开发者是否想得足够周到？行家从登录窗体上也能知道个一二。

例如有多个系统，这些系统是否支持单点登录？只用一套用户名密码可以登录多个系统？系统登录是否可以进行灵活配置？离职人员的处理、临时授权用户的登录系统问题、一个职员的多个系统帐号功能、系统帐号的密码策略等等、输入错误密码时的登录限制等等都能体现出设计、开发人员的水平。

2.1 请用“登录”不要“登陆”

登录系统大家经常犯的一个错误是写成“登陆系统”，曾经向资深人士咨询过此事，都觉得写成“登录系统”才是正确的，所以建议大家用正确的写法“登录系统”。而不是打仗时军队**登陆岛屿**，很多人会忽视这个用词、有些细节的确是应该注意的，有错就改、马上就改。



2.2 系统登录

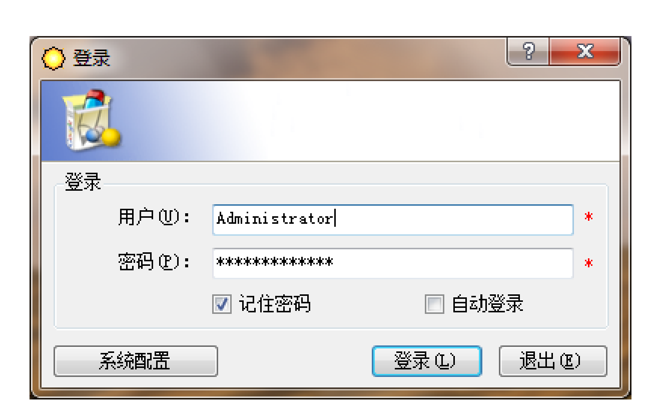


图 2-2-1 登录窗体

用户登录系统后，不管是什么帐号什么时候登录系统，都应该记录登录日志，就像平时管理严格的单位一样，有前台记录外来人员、出入门需要刷门禁系统一样，系统登录成功后应该更改帐号的在线状态，若登录失败的，都应该记录失败日志。输入的密码长度、允许输入错误密码的次数都应该进行限制，防止破坏分子用暴力方式猜测密码，需要对暴力猜测密码等行为设置障碍，防止入侵系统行为发生。



第3章 用户（帐号）管理

用户（帐号）管理模块主要由申请用户（帐号）、用户（帐号）审核、用户（帐号）编辑、创建用户（帐号）、设置用户（帐号）密码、设置用户（帐号）有效状态、删除用户（帐号）等等功能部分组成。



图 3-1 用户（帐号）管理菜单

创建新用户（帐号）可以由普通用户申请创建后由系统管理审核生效，也可以由系统管理员直接创建用户（帐号），多种创建用户（帐号）的方法可以减少系统管理员的工作量、可以更及时、准确的创建帐号，将工作量分摊到多个操作人员、管理员上，可明显提高工作效率。

通常情况下，新同事的用户（帐号），都由系统管理员进行录入创建生效，但是在一个大集团公司中有多个分公司的情况下或有异地办公机构时都由一个系统管理员录



入账号不只是效率高低的问题，由于没在现场工作出错的概率很高，若可以在任何分公司在任何一台电脑上进行申请用户（帐号）功能、甚至是直接在网上进行申请用户（帐号）步骤，相对来讲比单一系统管理直接创建用户（帐号）的做法工作效率会高一些，自己录入自己的帐号信息或者由身边的同事帮助录入信息会更准确一些、及时一些。

自己录入自己的帐号信息优点在于想用什么登录名就用什么登录名、这样在日常办公中使用信息系统也会更符合习惯一些、更贴切一些、也不用记过多的用户密码、至少可以使用自己多年习惯的用户名密码来登录业务系统。当然系统中的用户名是不可以重复的，注册时系统会自动检测是否数据已重复，若是由系统管理员创建的用户名不符合操作者的习惯又不允许用户自己可以自由修改登录用户名是一个很糟糕的事情、很恼火的事情。

在比较庞大的组织机构里有分公司有异地办公机构，使用本功能模块，还有分级审核、分级管理功能比较好，就是由某个分公司申请的账号可以由这个分公司的系统管理员进行审核管理，这个帐号的权限也可以由相应的管理员进行管理等，把整个系统管理员的工作可以分摊给各个分公司的系统管理员或者授权的操作员，可以更准确掌握实际工作情况、可以更加贴近实际权限配置、管理等，同时又能支持集中式的管理模式。

3.1 申请用户（帐号）

申请用户（帐号）模块主要是为了灵活、快速申请新帐号功能，还没分配到系统账号的用户可这个功能模块里提交申请用户（帐号）的信心，由系统管理员审核后，才可以登录业务系统。

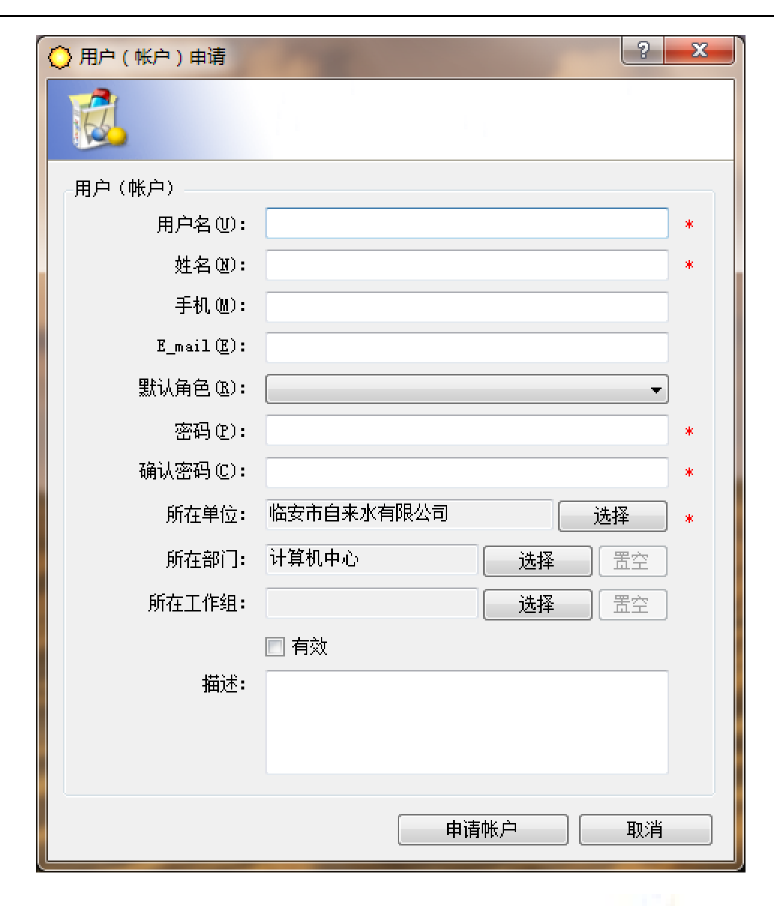
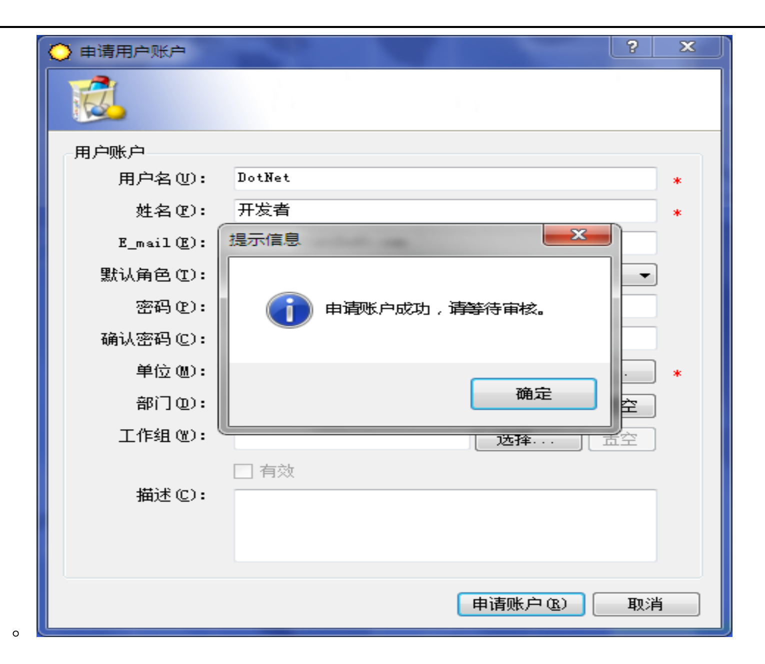


图 3-1-1 申请用户（帐号）窗体

准确填写好需要申请的用户名、密码、选择好所在的部门等相关信息，点击“申请帐号”按钮即可进入申请步骤，非管理员申请的用户（帐号）需要审核过程，在申请帐号的同时，会给相应的系统管理员发送即时信息，当然在管理员离线状态下时也可以通过邮件提醒方式或者手机短信提醒模式提醒管理者及时进行审核工作。

若需要有审核步骤，只有系统管理员进行审核通过后，所申请的账号会生效后才可以登录到相应的管理系统中



。

图 3-1-2 申请用户成功后的效果

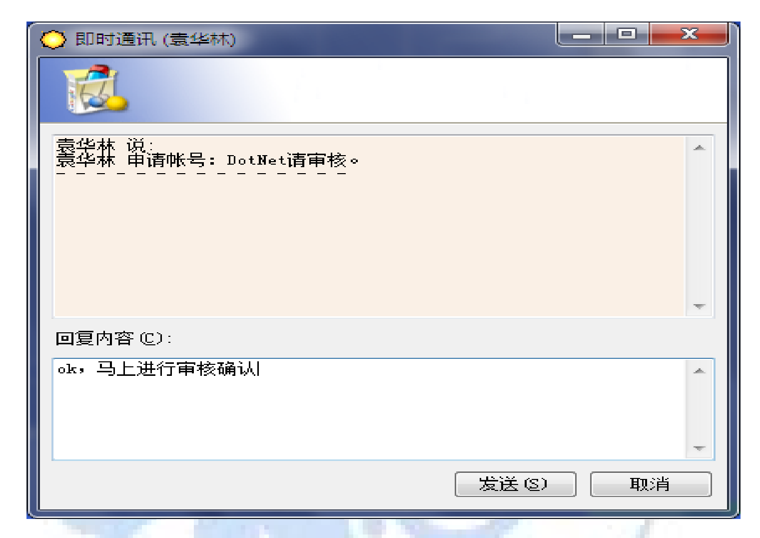


图 3-1-3 系统管理登录后的提醒信息效果

若系统中有未审核的帐号，系统会给系统管理员发送在线即时消息进行提醒，管理员可以在第一时间进行审核确认工作，主动提醒可提高工作效率，若无待审核帐号，也没必多余的操作，这样会提高系统的友善性，使系统变得更加人性化、智能化。

3.2 用户（帐号）审核

进入用户（帐号）审核功能页面，默认情况下、待审核的用户（帐号）数据会列

出来，系统管理员直接可以进行审核步骤，若整个系统是按分级管理方案部署的，那只会列出来当前管理员需要的数据列表出来，不是他所负责的待审核数据不会列出来。

可以通过帐号审核的查询功进行模糊查询数据、可以按审核状态、默认角色等条件进行过滤查询数据，可以更准确、快速查询数据。



图 3-2-1 用户（帐号）审核窗体

编辑功能可以对申请创建用户账号时填写信息进行修改，包括用户名、姓名、默认角色等。

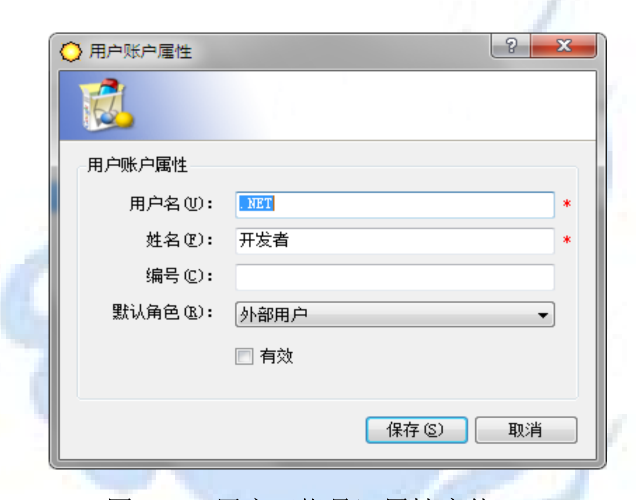


图 3-2-2 用户（帐号）属性窗体

用户（帐号）审核通过、驳回操作时，按注册时填写的邮件地址给用户发送提醒通知邮件，及时通知用户审核结果，审核通过的用户可以及时登录系统进行相应的业务操作，有邮件提醒功能可以提高工作效率、也不用打电话通知。

导出功能是将当前窗体中的数据导出为 Excel 格式的数据保存到本地计算机上，



方便用户进行二次数据处理。

删除功能，选择好需要删除的数据，点删除按钮即可，系统中被操作员误删的数据可以由数据库管理员进行恢复操作，应用系统里未进行物理删除数据、只对相应的数据打了删除标记处理，所以有误操作数据后恢复数据的可行性。



图 3-2-3 用户（帐号）审核窗体

修改用户（帐号）时不允许重复用户名，若输入了编号，编号不允许重复，系统会自动会检测数据是否已经重复，已被设置为删除标志的用户（帐号）不再重复验证的范围内，系统中的登录用户名也可以由管理员进行修改。用户（帐号）生效后，用户也可以自由修改用户（帐号），这样更符合用户的习惯，用户的排斥心里也会少很多。

给每个新用户及时分配个合适的编号，在真实的工作实际情况下，比较难做得到，所以建议可以先不输入用户编号，等用户的编号被确定下来后可以事后进行设置比较符合实际工作习惯。

3.3 用户（帐号）管理

用户（帐号）管理模块是系统管理员最常用的功能模块之一，在这个模块里可以直接创建用户（帐号）、修改用户（帐号）信息、可以设置用户的密码。

用户（帐号）管理功能也能实现分级管理功能，例如每个分公司管理每个分公司的帐号、由超级管理员管理所有的用户（帐号）功能。

员工离职时不建议直接删除用户（帐号），而是停用用户（帐号），例如员工离职了，很多系统中的历史真实数据记录不应该被删除，员工离职时可能最后一个月的工资还没发放，相关的离职手续也没办理好，应该将用户设置为停用状态（取消有效状态），用户（帐号）被随便删除后有可能会引起系统中其他相关数据出现关联错误的隐患。

员工离职后，就把帐号删除了，可能会引发一些数据关联性、引用上的错误，例如人辞职了，以前的工资数据、工作日志数据、项目数据等等都已是发生过的真实数据，不应该把这个用户给删除了，相关数据都应该作为历史数据保留下来、留下历史痕迹才是正确的做法。

按道理，一切数据都不应该进行物理删除，而是进行逻辑删除才对，例如进行了删除操作，只是给数据打上一个删除标志（例如数据表里的 DeleteMark 字段标志设置为1、表示已经被删除了），若真需要删除数据，应该由数据库管理员进行真正的物理删除操作，物理删除一条数据是非常容易，但是由于误删了一条数据，想把这条数据再恢复过来的代价往往是非常昂贵的，可能要花费 10 倍、100 倍的代价才能把这个数据恢复过来，就是提醒了用户是否确认删除？用户也按了确认也不应该进行物理删除操作。





图 3-3-1 用户（帐号）管理窗体

添加新用户（帐号）的效果图如下，当然这个窗体也可以进行二次开发，按自己的需要增加字段等等，例如把手机号码、电子邮件地址也录入进来或进行一些其他的相关连的事务处理。

系统管理员帐号不应该轻易的被删除、用户名、编号等不应该重复的检查，当然不设置编号也是可以的，新来的员工一般都没有明确的编号，可以事后再补充也是很常见的，新建帐号时必须要输入编号，而且必须输入个不重复的编号相对友善度不高、属于系统缺乏人性化、考虑不周全。

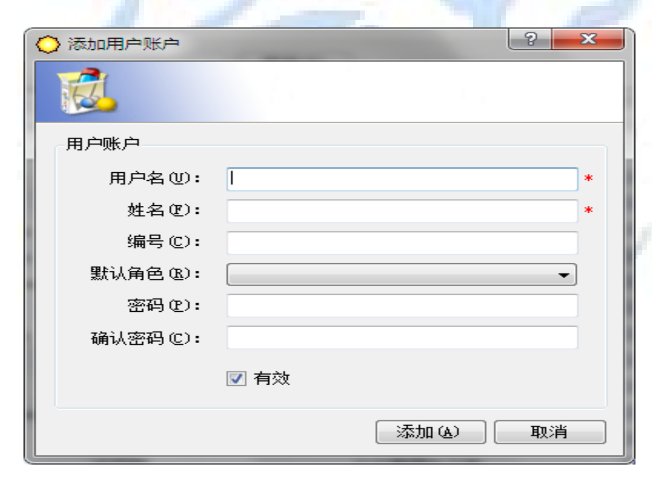


图 3-3-2 添加用户（帐号）窗体

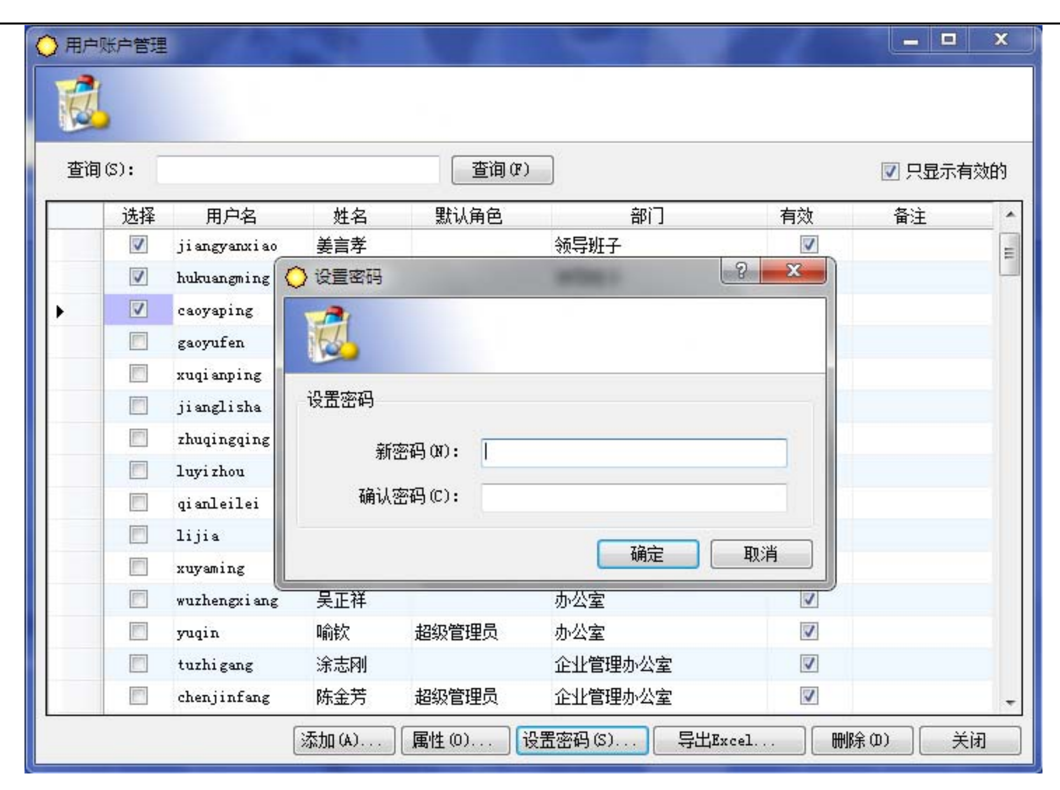


图 3-3-3 设置密码窗体



图 3-3-4 用户（帐号）管理窗体

用户（帐号）的导出功能，主要是为了以下几个工作需要，你可能做的是一个会员管理系统，定期需要给会员发送手机短信、宣传邮件提醒等功能，你已经购买了相应

的软件或者服务，经常需要把数据导出后，快速导入到相应的另外软件系统里，达到你数据共享的目的。同时定期导出数据到其他文件中也起了一个备份数据的用途，用户的数据是宝贵的资源，这些数据将会产生与公司相关连的业务，资源极其重要，所以这部分数据的备份工作意识需要强一些，客户的数据都丢了，怎么联系客户、怎么保障稳定的收入呢。

当然数据导出功能也可以按客户的个性化需要，进行二次开发，按自己的实际需求修改源码或者做一些接口处理等等，甚至直接可以到数据库中进行处理，因为相应的表结构等都有明确的定义。

有些用户的帐号又不能失效、又不想让这个用户登录，那可以由管理员来设置这个帐号的密码，这样这个用户就不能登录到信息系统里来了，也不用修改无效状态就可以了。

系统中的用户名是允许修改的，这一点很多系统做得都有些不友善，例如系统管理员分配了一个不喜欢的帐号名，想修改也改不了，就想改，也麻烦一些，设置是需要一个过程等等，所以自己愿意用什么用户就用什么用户名还是很有必要的，当然系统管理员也应该能干涉，能管理是非常必要的。

3.4 分布式管理

每个公司管理每个公司自己的数据，日常生活中经常会遇到需要管理每个公司自己的数据，然后由集团总公司集中管理所有数据的需求。

我们经常会开发这个系统，那个系统，每个系统经过不断完善优化后我们才会发现，刚开始都是需要很简单的系统，但是往往到最后都会变得很复杂，最终我们还是逃脱不了重复建设。

那我们还不如刚开始的时候，就用一个考虑的比较充分的配置管理工具来管理我们的 A 系统、B 系统，用这个系统可以配置管理非常复杂的权限配置，不管将来客户的



需求怎么变来变去，都能适应，而且不用费什么力气，配置来配置去就可以了，当然一些简单的权限，当然是更容易能达到目的了。

很多时候，我们开发人员开发出来的某个管理系统，往往经不客户的折腾，最后改来改去会变得异常脆弱，要么这里出错那里出错，要么这里不稳定要么那里不稳定，很多时间都耗费在基础性的建设工作上，这时候我们也需要一个铜墙铁壁的基础组件，用这个组件来配置管理整个系统的权限，用户，角色等就可以了。

本着把简单的软件做个彻底，彻底的铜墙铁壁，没有错误，没有漏洞，可以重复利用劳动成果的指导思想继续完善几年打造完美的精品。

铜墙铁壁的基础数据是一切管理系统及多个管理系统之间进行整合的核心基础，没有稳定的基础数据就没稳定的业务管理管理系统，所以我们开发任何一个管理系统之前就应该准备好铜墙铁壁的基础组件，可以灵活配置管理扩展的“软件数据中心”，而且是源码全部公开的、数据库结构设计全部公开的核心组件。

1：首先我们应该定义一个操作权限。

我们先登录到系统里

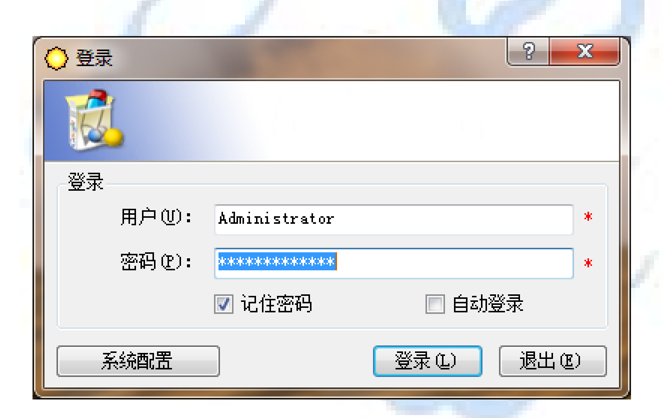


图 3-4-1 用户（帐号）登录窗体

再选操作权限项定义管理菜单进入模块

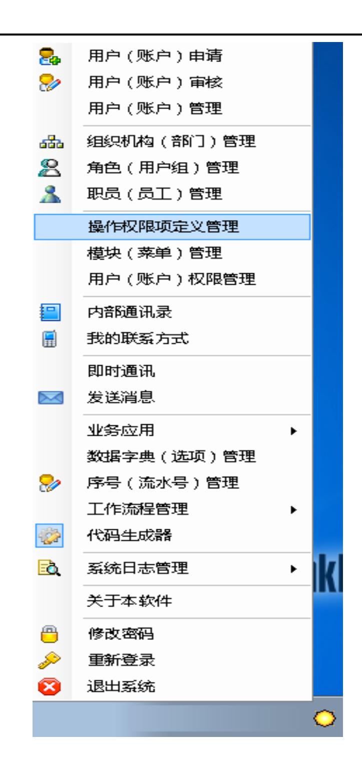


图 3-4-2 用户（帐号）管理菜单

在此模块里，定义了一个操作权限，叫"走火入魔权限配置后台管理"，然后把这个权限项定义为数据集权限，意思是设置这个权限时需要设置管理范围。

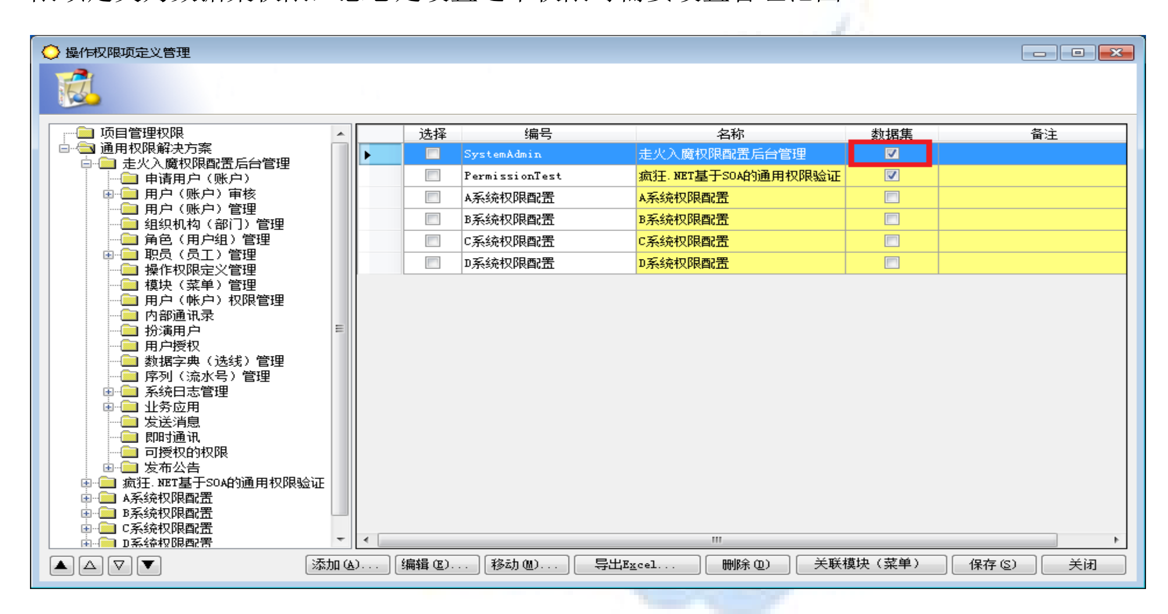


图 3-4-3 用户（帐号）权限管理

2：操作的用户需要有相应的权限并且需要设置权限范围，我们选择用户（帐号）权限管理菜单

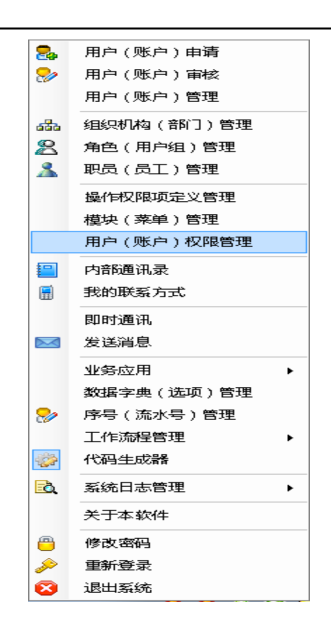


图 3-4-4 用户（帐号）管理菜单

接着选择目标用户帐号，对此帐号进行详细的权限设置。

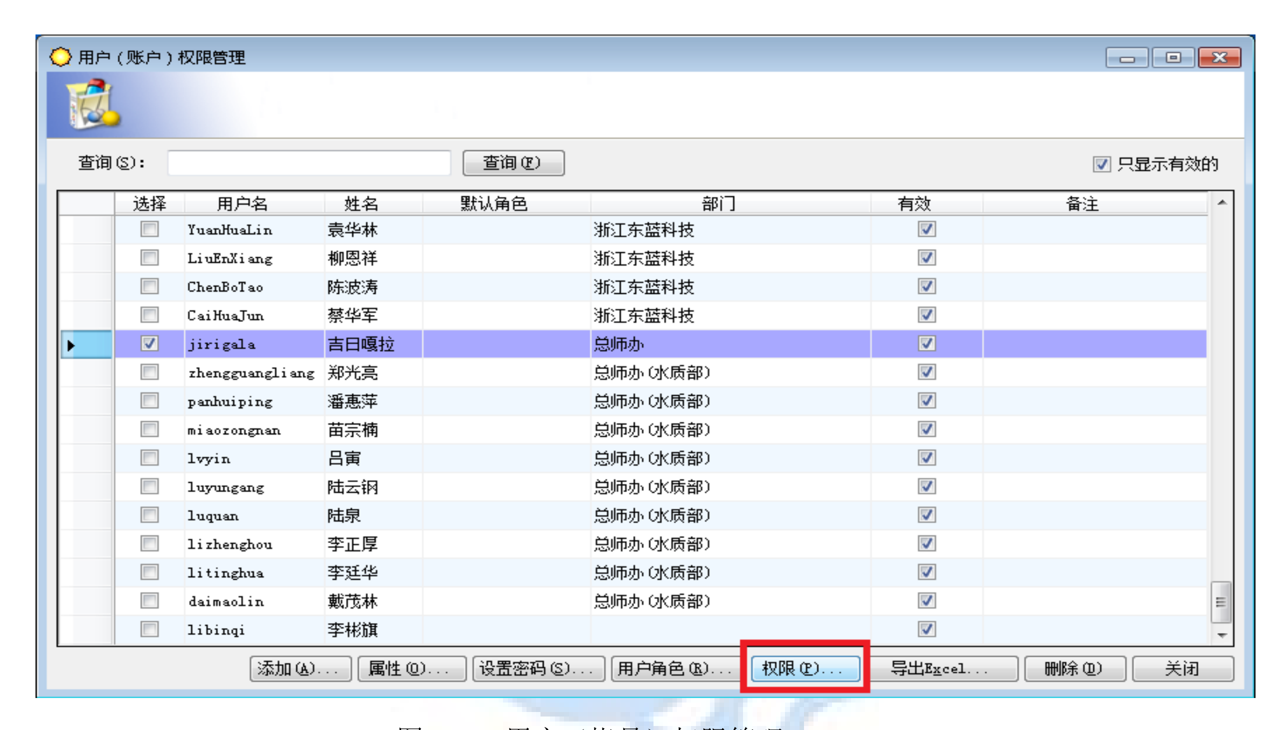


图 3-4-5 用户（帐号）权限管理

给目标用户，权限后台管理配置的权限，并对此权限进行范围设置（数据集权限设置）。

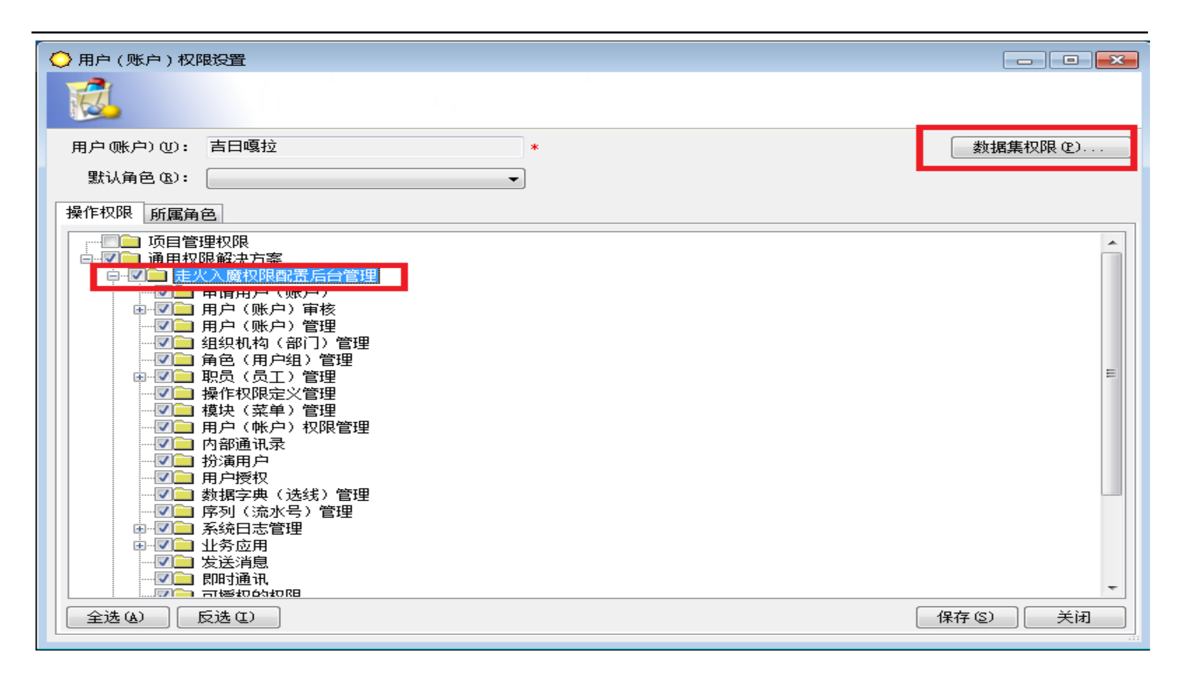


图 3-4-6 用户（帐号）权限管理

当前用户的操作权限范围，可以进行按明细设置，选择详细的权限范围，到此已正确设置帐号的权限范围。

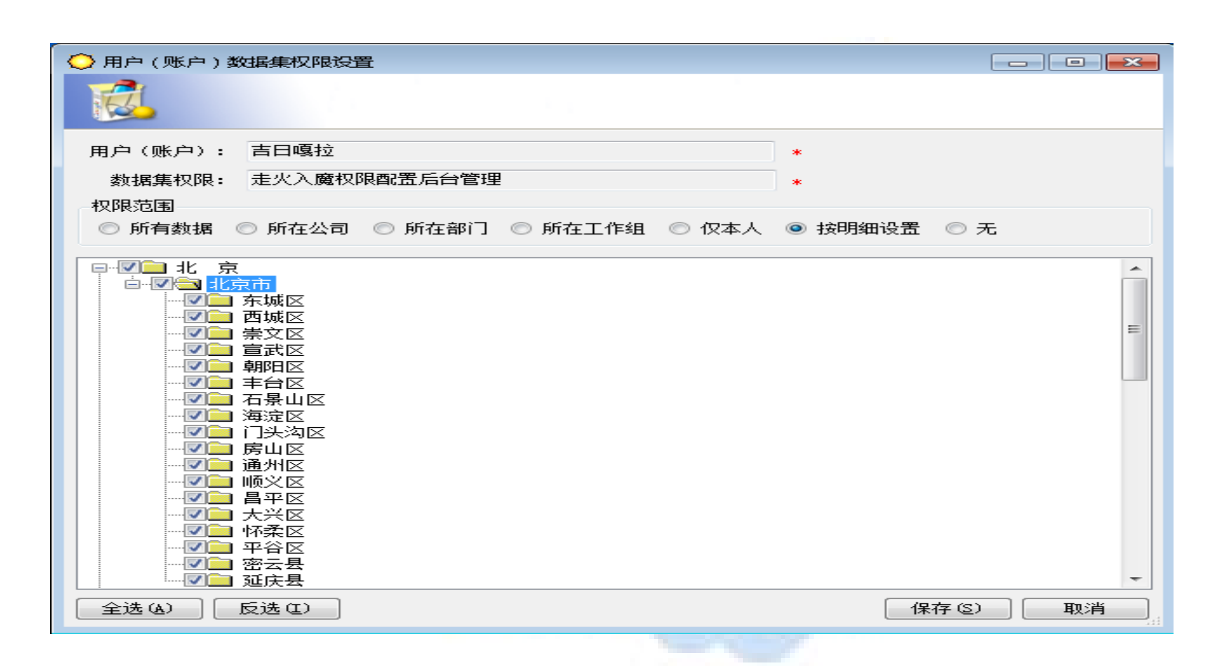


图 3-4-7 用户（帐号）数据集权限管理

3：输入演示数据。

我们先模拟输入一个北京的用户帐号申请情况。

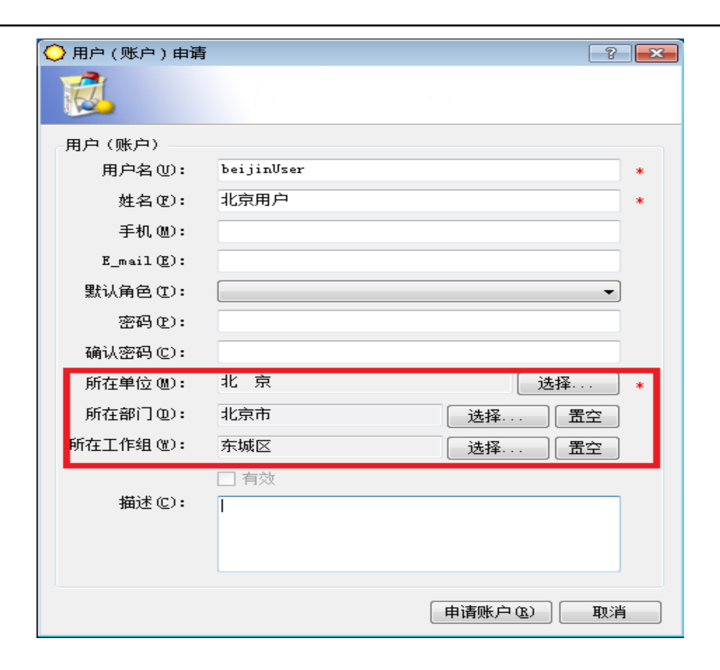


图 3-4-8 用户（帐号）申请

我们再输入一个上海的用户帐号申请情况。

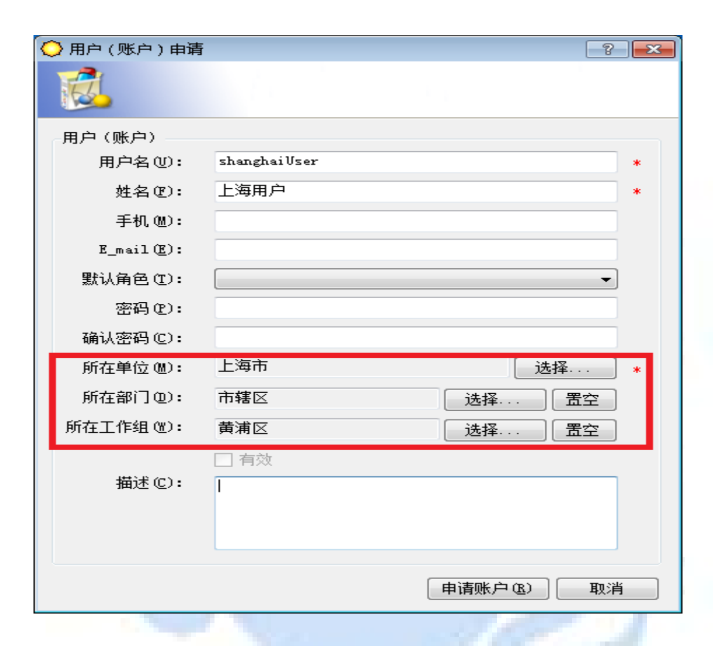


图 3-4-9 用户（帐号）申请

我们用系统管理员进入审核模块看看，是否有 2 个帐号是待审核状态？

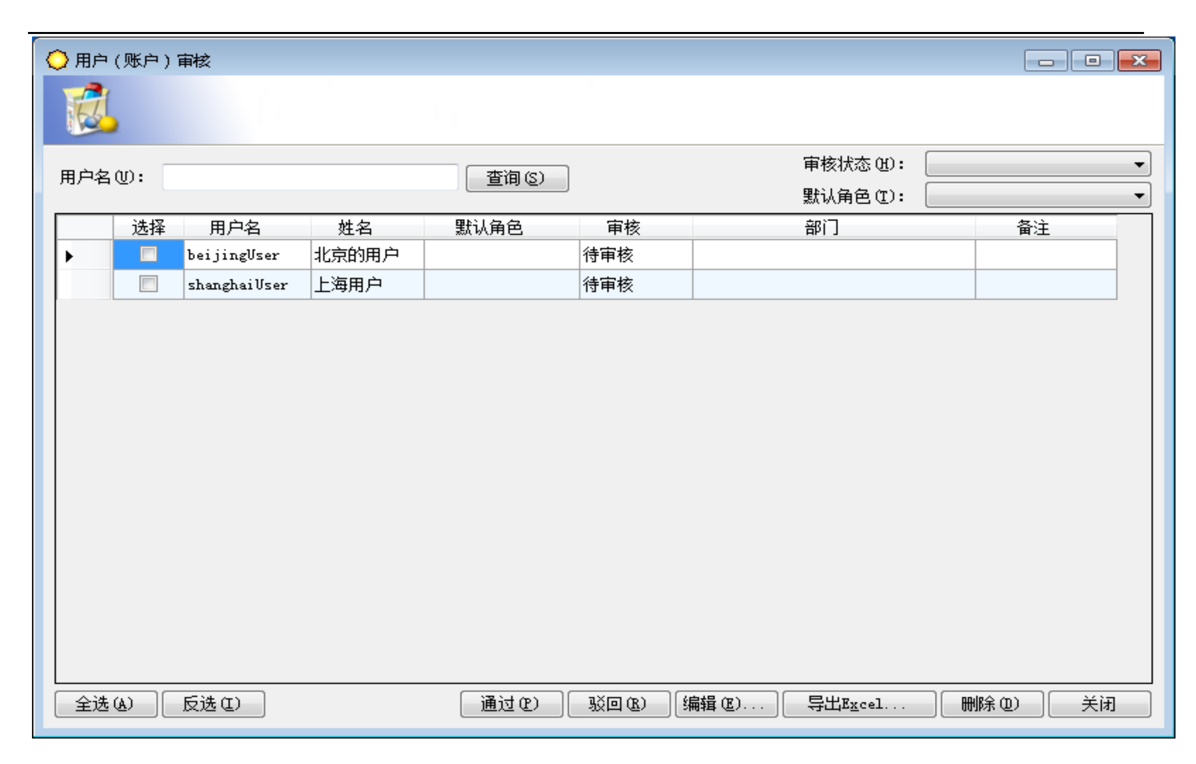


图 3-4-10 用户（帐号）审核

4：是否达到了每个分公司管理每个分公司数据的目的。

我们用目标帐号"吉日嘎拉"登录此权限配置管理后台，

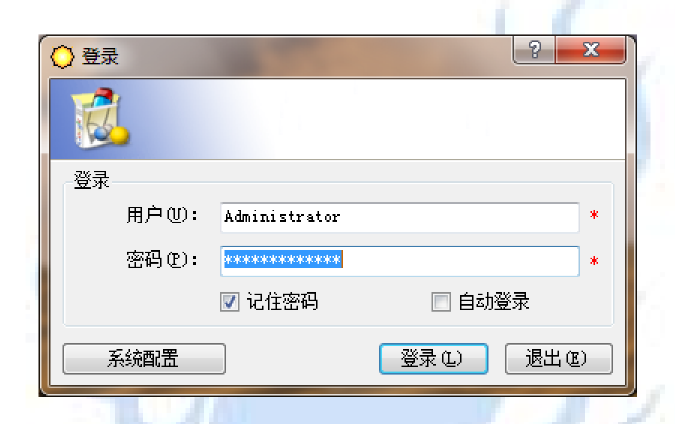


图 3-4-11 用户（帐号）登录

进入用户帐号审核模块，看看会是什么情况？



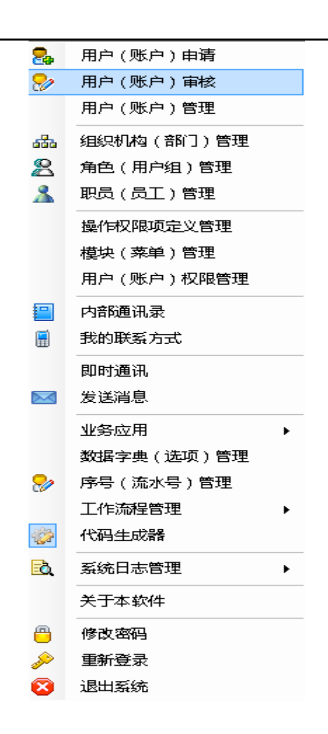


图 3-4-12 用户（帐号）管理菜单

的确按我们的数据集权限设置范围，只出现了北京的用户审核数据，上海的帐号申请数据没出现在当前操作窗体里。

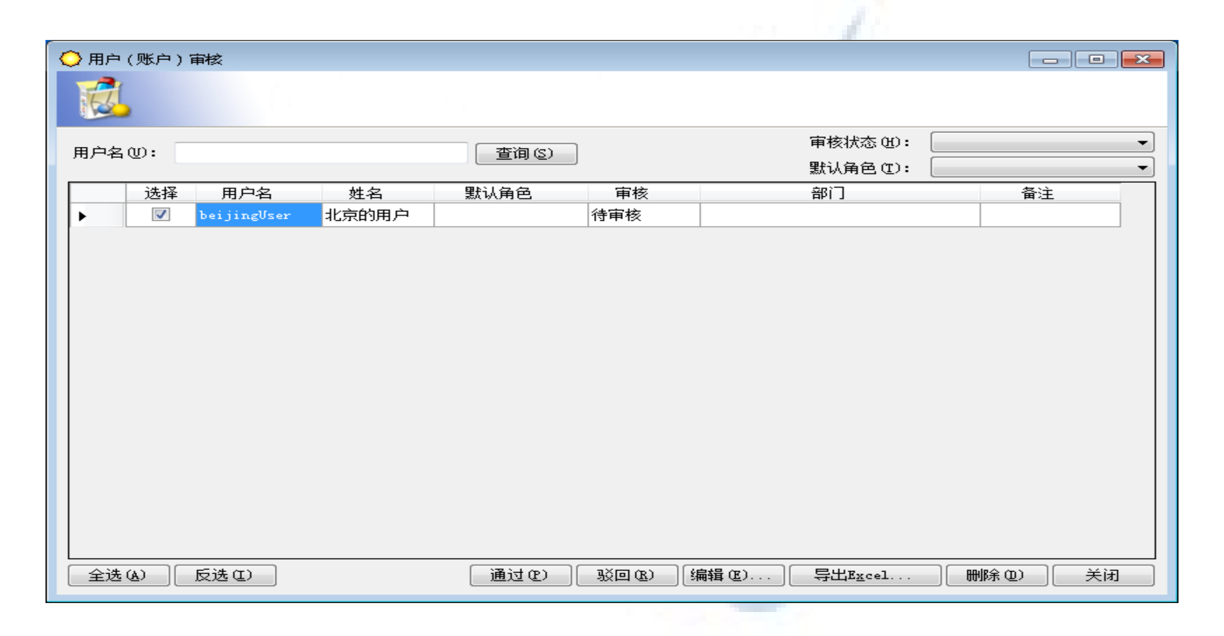


图 3-4-13 用户（帐号）审核

从数据设计的思想来讲，非常简单的。

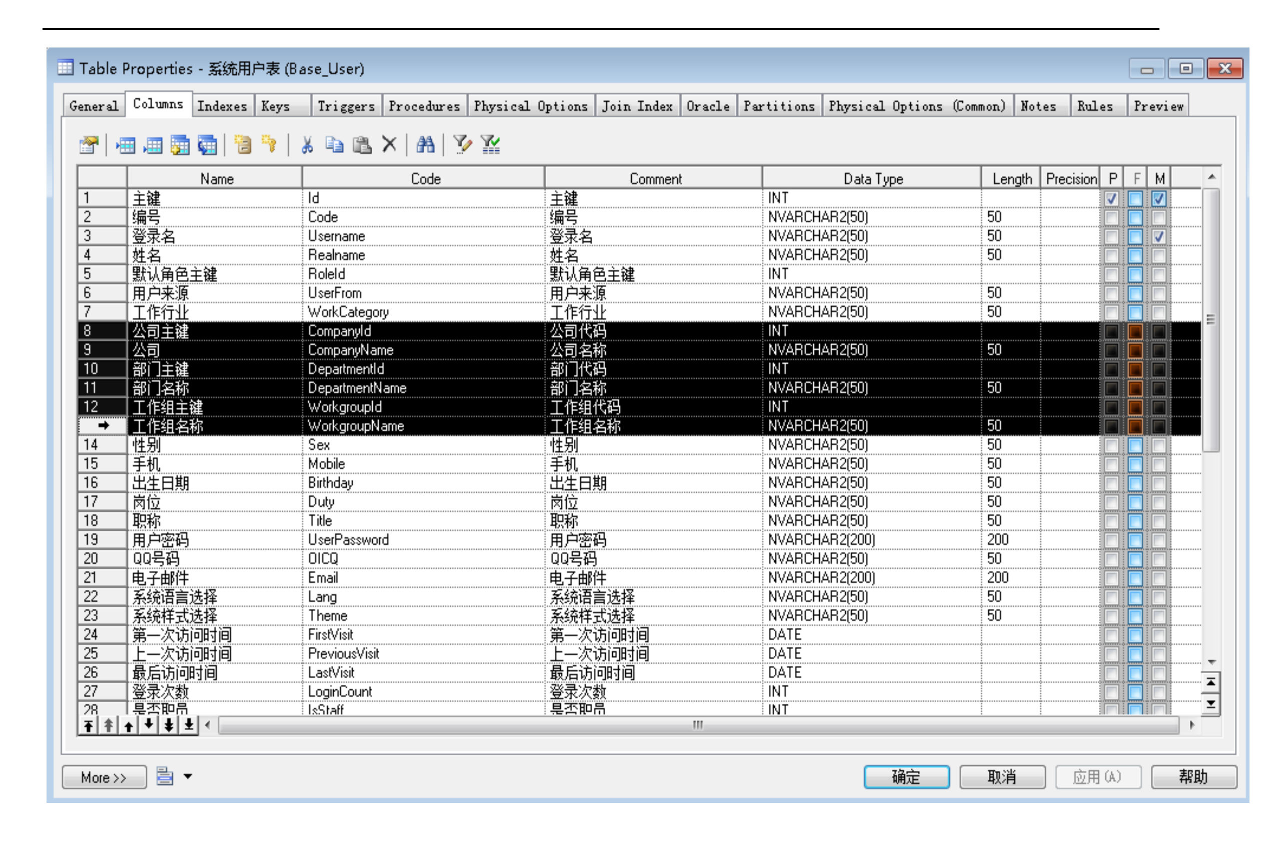


Table Properties - 系统用户表 (Base_User)

General | Columns | Indexes | Keys | Triggers | Procedures | Physical Options | Join Index | Oracle | Partitions | Physical Options (Common) | Notes | Rules | Preview

	Name	Code	Comment	Data Type	Length	Precision
1	主键	Id	主键	INT		
2	编号	Code	编号	NVARCHAR2(50)	50	
3	登录名	Username	登录名	NVARCHAR2(50)	50	
4	姓名	Realname	姓名	NVARCHAR2(50)	50	
5	默认角色主键	RoleId	默认角色主键	INT		
6	用户来源	UserFrom	用户来源	NVARCHAR2(50)	50	
7	工作行业	WorkCategory	工作行业	NVARCHAR2(50)	50	
8	公司主键	CompanyId	公司代码	INT		
9	公司	CompanyName	公司名称	NVARCHAR2(50)	50	
10	部门主键	DepartmentId	部门代码	INT		
11	部门名称	DepartmentName	部门名称	NVARCHAR2(50)	50	
12	工作组主键	WorkgroupId	工作组代码	INT		
→	工作组名称	WorkgroupName	工作组名称	NVARCHAR2(50)	50	
14	性别	Sex	性别	NVARCHAR2(50)	50	
15	手机	Mobile	手机	NVARCHAR2(50)	50	
16	出生日期	Birthday	出生日期	NVARCHAR2(50)	50	
17	岗位	Duty	岗位	NVARCHAR2(50)	50	
18	职称	Title	职称	NVARCHAR2(50)	50	
19	用户密码	UserPassword	用户密码	NVARCHAR2(200)	200	
20	QQ号码	OICQ	QQ号码	NVARCHAR2(50)	50	
21	电子邮件	Email	电子邮件	NVARCHAR2(200)	200	
22	系统语言选择	Lang	系统语言选择	NVARCHAR2(50)	50	
23	系统样式选择	Theme	系统样式选择	NVARCHAR2(50)	50	
24	第一次访问时间	FirstVisit	第一次访问时间	DATE		
25	上一次访问时间	PreviousVisit	上一次访问时间	DATE		
26	最后访问时间	LastVisit	最后访问时间	DATE		
27	登录次数	LoginCount	登录次数	INT		
28	是否职员	IsStaff	是否职员	INT		

More >> 确定 取消 应用(A) 帮助

图 3-4-14 用户（帐号）列表

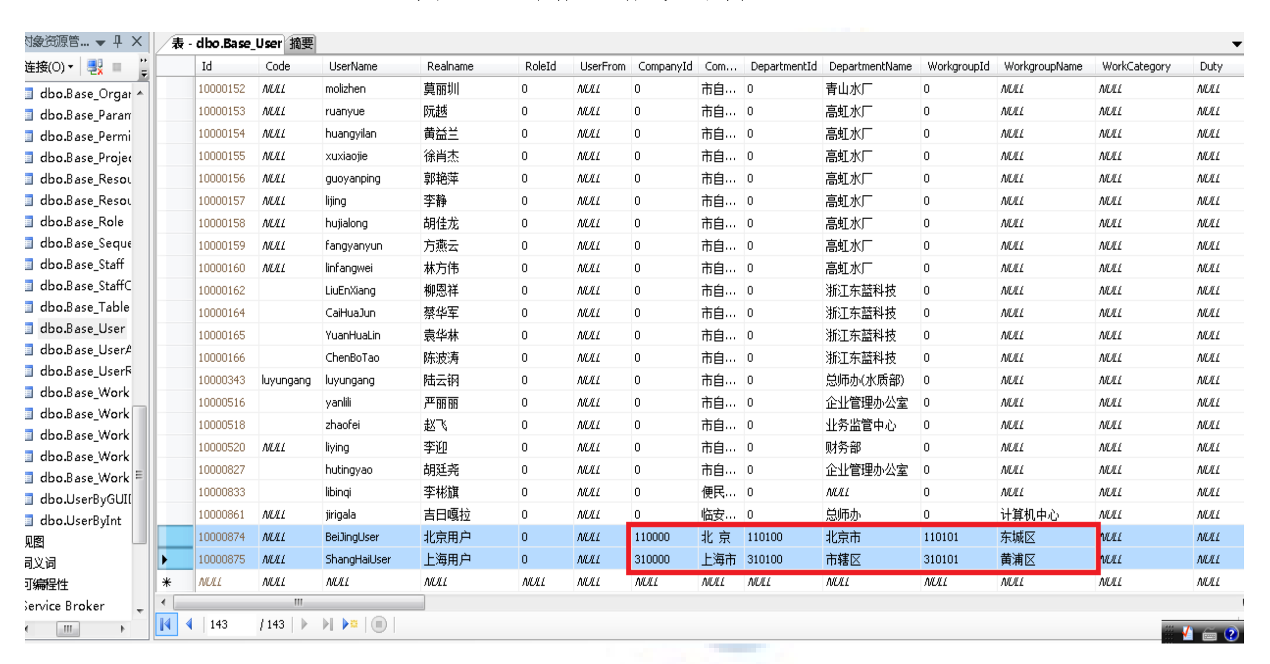


表 - dbo.Base_User 摘要

Id	Code	UserName	Realname	RoleId	UserFrom	CompanyId	Com...	DepartmentId	DepartmentName	WorkgroupId	WorkgroupName	WorkCategory	Duty
10000152	NULL	molizhen	莫丽圳	0	NULL	0	市自...	0	青山水厂	0	NULL	NULL	NULL
10000153	NULL	ruanyue	阮越	0	NULL	0	市自...	0	高虹水厂	0	NULL	NULL	NULL
10000154	NULL	huangyilan	黄益兰	0	NULL	0	市自...	0	高虹水厂	0	NULL	NULL	NULL
10000155	NULL	xuxiaojie	徐肖杰	0	NULL	0	市自...	0	高虹水厂	0	NULL	NULL	NULL
10000156	NULL	guoyanping	郭艳萍	0	NULL	0	市自...	0	高虹水厂	0	NULL	NULL	NULL
10000157	NULL	lijing	李静	0	NULL	0	市自...	0	高虹水厂	0	NULL	NULL	NULL
10000158	NULL	hujialong	胡佳龙	0	NULL	0	市自...	0	高虹水厂	0	NULL	NULL	NULL
10000159	NULL	fangyanyun	方燕云	0	NULL	0	市自...	0	高虹水厂	0	NULL	NULL	NULL
10000160	NULL	linfangwei	林方伟	0	NULL	0	市自...	0	高虹水厂	0	NULL	NULL	NULL
10000162		LiuEnXiang	柳恩祥	0	NULL	0	市自...	0	浙江东蓝科技	0	NULL	NULL	NULL
10000164		CaiHuaJun	蔡华军	0	NULL	0	市自...	0	浙江东蓝科技	0	NULL	NULL	NULL
10000165		YuanHuaLin	袁华林	0	NULL	0	市自...	0	浙江东蓝科技	0	NULL	NULL	NULL
10000166		ChenBoTao	陈波涛	0	NULL	0	市自...	0	浙江东蓝科技	0	NULL	NULL	NULL
10000343	luyungang	luyungang	陆云钢	0	NULL	0	市自...	0	总师办(水质部)	0	NULL	NULL	NULL
10000516		yanlili	严丽丽	0	NULL	0	市自...	0	企业管理办公室	0	NULL	NULL	NULL
10000518		zhaofei	赵飞	0	NULL	0	市自...	0	业务监管中心	0	NULL	NULL	NULL
10000520	NULL	liying	李迎	0	NULL	0	市自...	0	财务部	0	NULL	NULL	NULL
10000827		hutingyao	胡廷尧	0	NULL	0	市自...	0	企业管理办公室	0	NULL	NULL	NULL
10000833		libinqi	李彬旗	0	NULL	0	便民...	0	NULL	0	NULL	NULL	NULL
10000861	NULL	jirigala	吉日嘎拉	0	NULL	0	临安...	0	总师办	0	计算机中心	NULL	NULL
10000874	NULL	BeiJingUser	北京用户	0	NULL	110000	北 京	110100	北京市	110101	东城区	NULL	NULL
10000875	NULL	ShangHaiUser	上海用户	0	NULL	310000	上海市	310100	市辖区	310101	黄浦区	NULL	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

143 / 143

图 3-4-15 用户（帐号）列表

第4章 组织机构（部门）管理

通俗的讲，就是一个集团公司，划分几个区域，每个区域都有什么分公司，每个分公司都有哪些部门一样，哪些部门又有那些子部门等。

当然也可以是外贸公司的全球性客户、合作伙伴的分布情况管理、或者是一个行政管理单位的按地区划分的行政机构的树型管理。

一般使用信息化系统的公司，大多是人数众多、地域分布相对广有异地分支机构的更迫切使用信息化管理系统，关键商业数据都需要依靠信息化技术及时进行沟通协调，提高办事效率。

软件系统需要充分考虑有多个分公司的情况下，应该如何满足管理上的需要，如何才能做得有条有理？这时就需要有组织机构管理的理念来引导系统架构，可以充分考虑哪个负责人可以管辖哪个区域的数据等，这样也防止没有经过授权的人越级查看不应该看的客户资料、合同信息等重要商业信息。

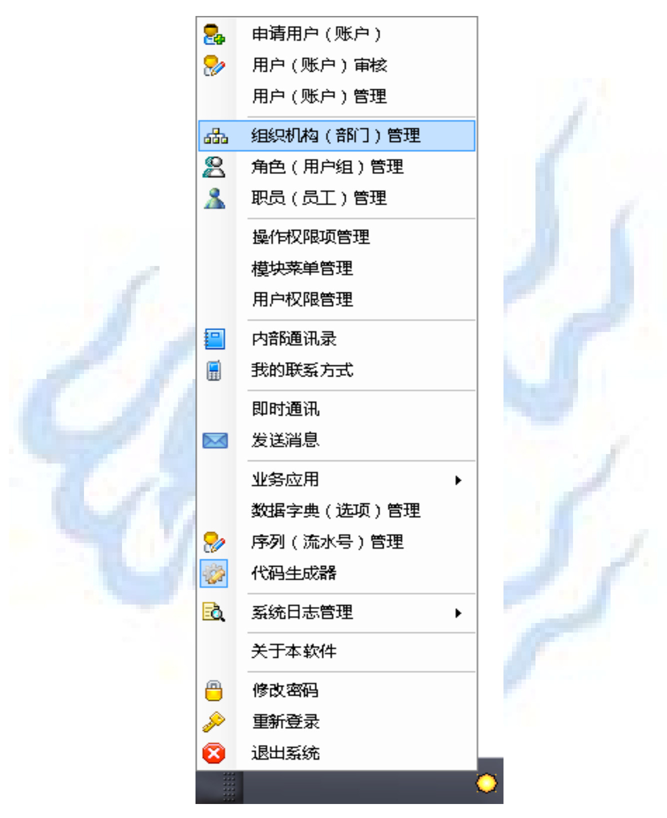


图 4-1 组织机构（部门）管理菜单

4.1 大型业务系统

属于大型集团公司，全国各地可能有多个分公司，可以划分区域、分公司等等，或者客户遍布于世界各地，需要严格的按区域划分、组织机构划分等进行严格的控制管理，每个管理员只在相应的管理范围内进行业务管理等。

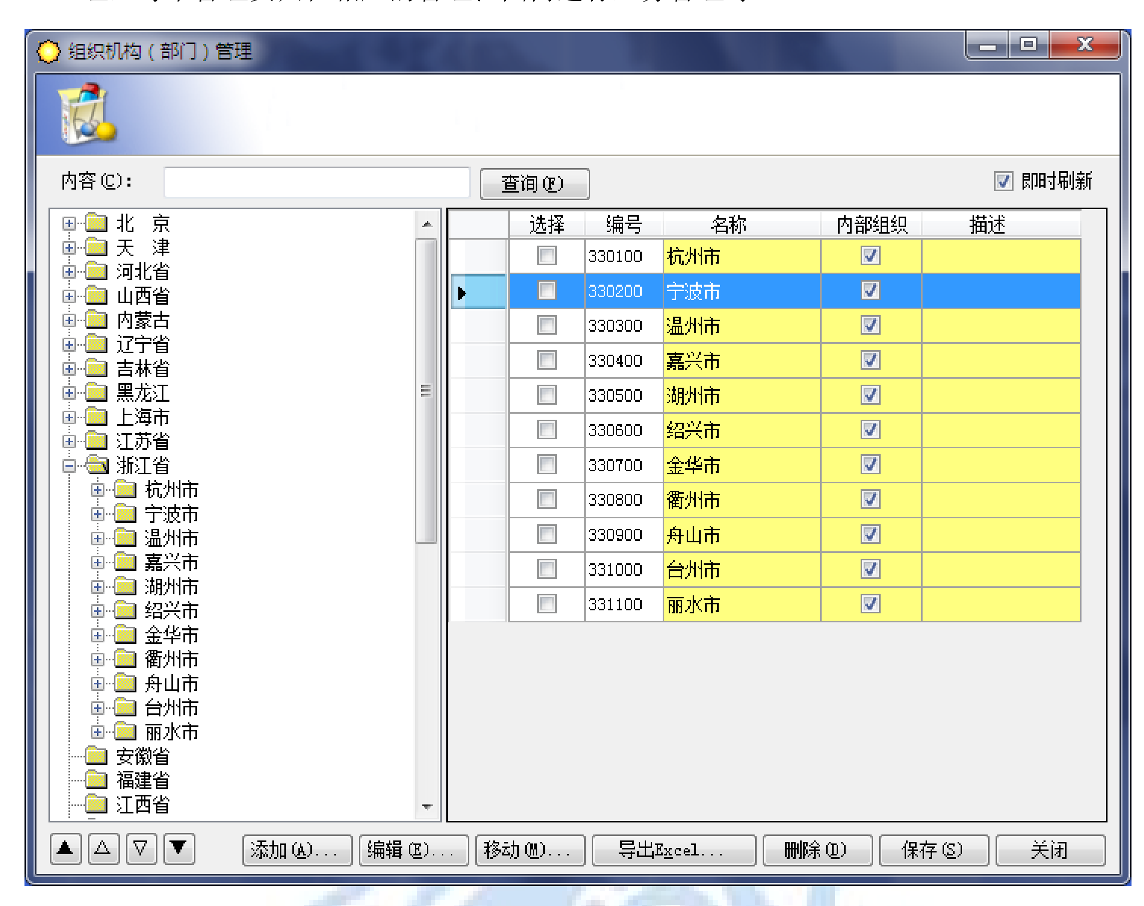


图 4-1-1 组织机构管理窗体

在大型的信息化管理系统中，往往需要分公司的系统管理员来进行管理各自权限范围内的数据，例如各分公司的用户（帐号）、各分公司的日常管理、权限配置维护等等，各分公司的系统管理员往往对所在公司的情况了解更及时更准确。

当然也可以按集中管理策略也可以，并不是非需要分公司系统管理，这要由公司的管理能力及办事效率而决定。

组织机构管理模块也有适当的整个公司的规划功能，到底设立哪些部门，各个部门都有哪些工作职能，人员配置情况，岗位角色规划等等。



4.2 中小型业务系统

一般一个上百人左右的中小型公司里所有内部员工都是用的软件系统，主要以公司内部管理系统为主，例如需要规划管理好公司内部的部门设置情况，人员归属情况，可以由系统管理员进行统一的管理，例如下图。

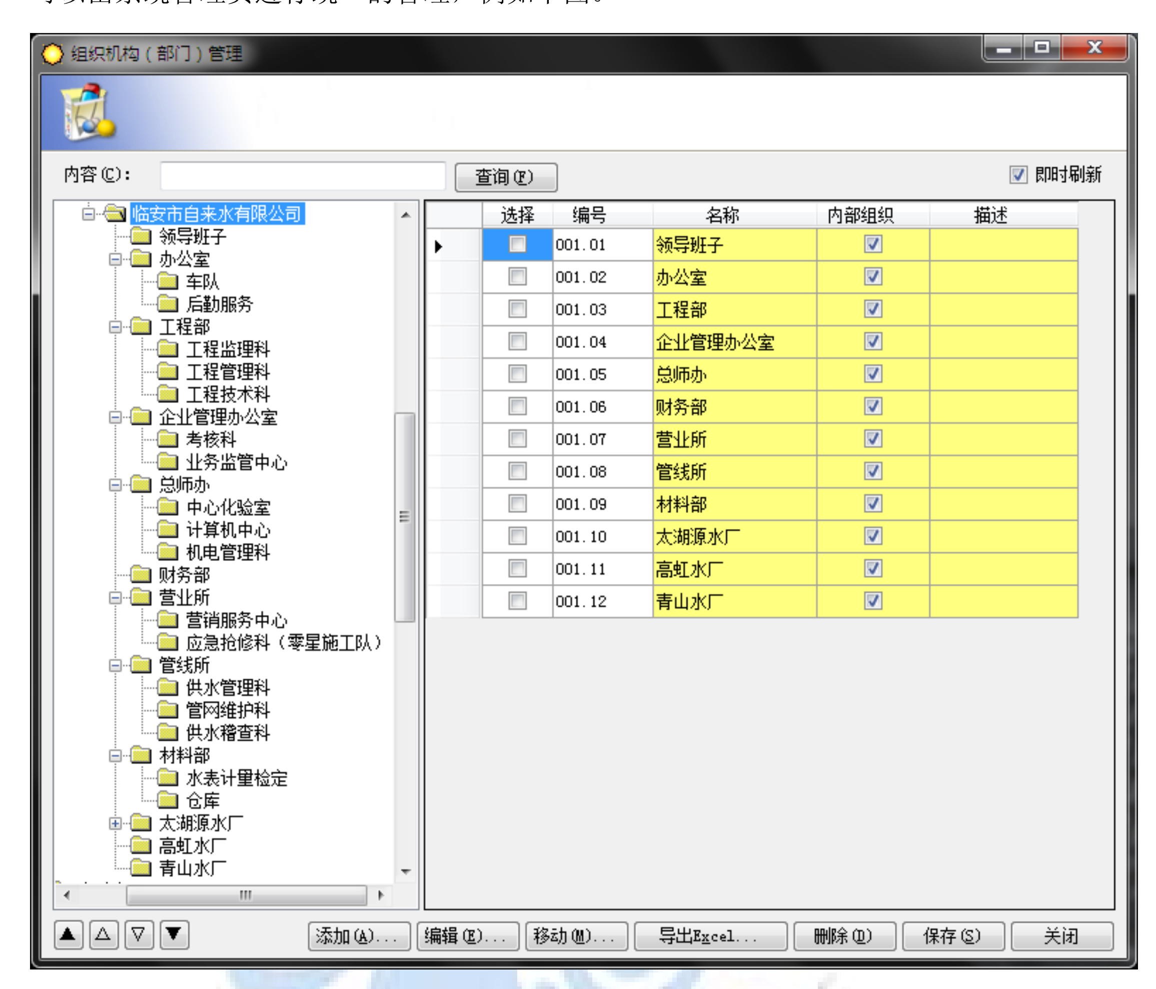


图 4-2-1 组织机构管理窗体

可以通过组织机构（部门）管理工具快速配置好公司的内部部门设置情况等，这样就可以在相应的业务功能模块采用上图中的数据，各个模块都可以有效利用部门的划分及相应的部门中的员工数据等。

组织机构的默认排序顺序是非常重要的，若每次显示出来的排序顺序都是乱的，那每天应用系统的人会很不爽，其实大家都有一个习惯，哪个部门排在哪个位置，他们的先后顺序都可以灵活设置比较好，这样可以更贴近客户的习惯，软件的友善性会更高一些。

4.3 微型的业务系统

有些很简单的业务系统，也不涉及到组织机构等，只要有几个用户，甚至就是一个用户，连角色什么也不需要，只要这个用户能登录系统，然后所有的事情都可以做就可以了，这时可以不用考虑那么多组织机构管理、角色权限管理等，就把这 2 个功能模块在模块管理中屏蔽掉就可以了。

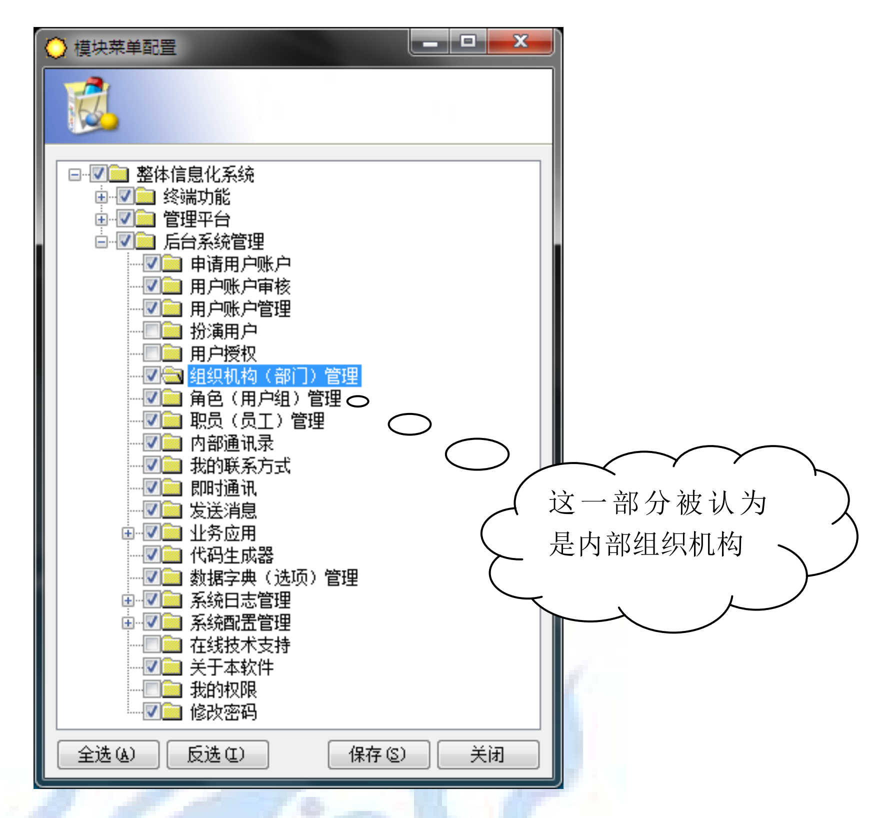


图 4-3-1 模块（菜单）配置窗体

那一个用户什么都可以操作，怎么处理比较好呢，首先可以建议使用 Administrator 这个登录名，这个是超级管理员的意思，只要用了这个帐号，就是所有的权限都不用配置，就应该有所有的操作权限，或者还可以将用户归于到 Administrators 角色中，这个角色是超级管理员的意思，任何一个帐号被归到这个角色中，也有所有系统的操作权限等，是系统中权限最大的用户、角色。

小公司也有发展壮大的可能性、系统里多几个空表，可扩展性更加良好，为将来扩展起来更好扩展，也不用非要删除了。那么几个表，也不会咬，若多余的功能菜单觉得很碍眼，那可以设置为无效状态，也不会在管理工具里显示出来了，购买笔记本电脑，

也不会因为不需要无线网络，然后要求厂商把无线网络模块去掉的吧，说不定哪天就派上用处了。

4.4 内外部组织机构

组织机构可以分为内部组织机构与外部组织机构，例如，客户的分布区域属于外部组织机构，一个公司的内部的部门的划分等，属于内部组织机构部分，例如下图所示。

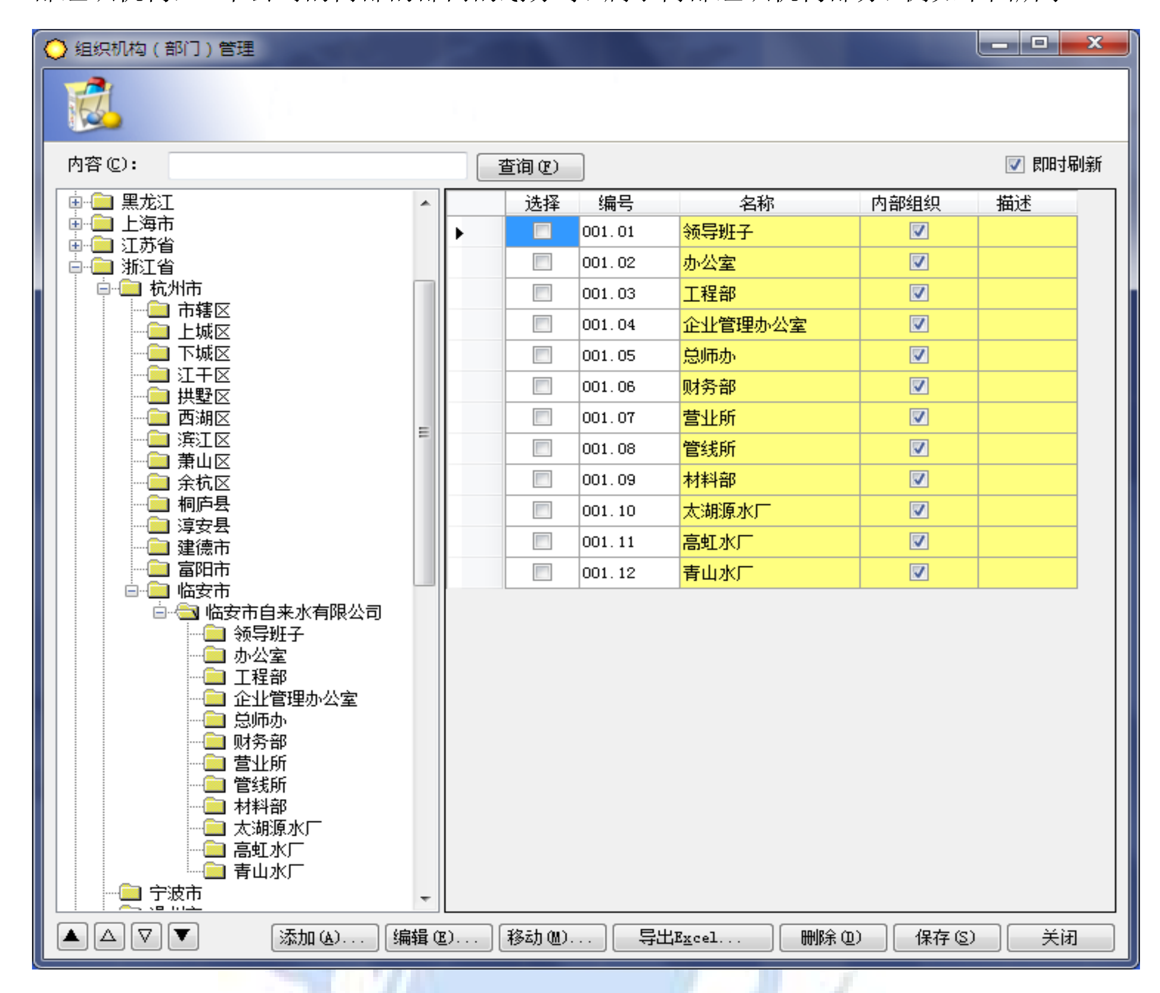


图 4-4-1 组织机构管理窗体

这样做的好处是，组织机构的管理做一套就可以了，不用内部组织机构的管理做一个，外部组织机构的管理又做一套程序，这样就可以集中管理所有的组织机构了，将来由于业务的拓展等，需要数据集权限时，对整个系统的影响会很小，也留下了将来深入二次开发的余地，这个系统数据的条理性也会更强一些。

虽然现在没有严格的数据过滤需要，将来随着业务的深入、公司规模的发展，也

有可能在不久的将来，客户就需要按组织机构进行数据库过滤的实际工作及管理上的迫切需要了。



第5章 角色（用户组）管理

5.1 角色和用户组

此功能模块可以灵活设置整个系统内的角色/用户组信息，可以进行对角色的添加、修改、删除、分配权限的操作，可以灵活设置某个角色中包含的用户（帐号）等。

角色：类似岗位定义、主要是为了系统分配权限、定义工作流程中设置审批节点，提高多用户（帐号）的权限分配效率、减少用于帐号的变动引起的重复设置权限的工作量等。

用户组：临时性的用户组或由系统用户创建的用户组，可以自由定义，可以定位为自己管理自己所创建的用户组。

此功能模块主要是为了解决如下几个问题:

1: 系统都有哪些角色/用户组?

2: 哪些角色/用户组,有哪些成员?

3: 哪些角色/用户组,有什么权限?

4: 谁可以对角色/用户组进行管理? 设置权限、管理其中的用户?

5: 谁能管理哪些角色/用户组

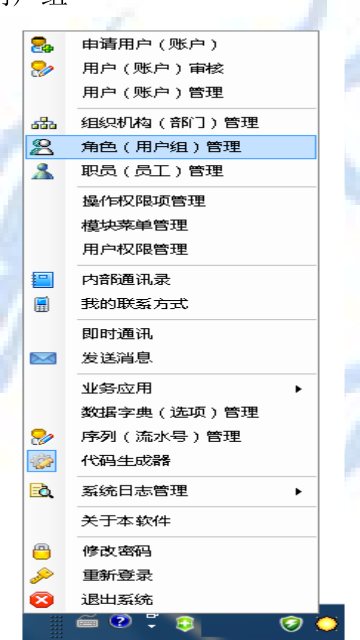


图 5-1 角色（用户组）管理菜单

在查询内容中输入需要查询的信息，系统会根据输入的查询信息显示相关的角色信息，添加的角色编号、名称不能重复，超级管理员角色不能被删除、不能被更改，是系统默认的角色，角色可以进行排序工作、停用设置、删除、导出等等。

可以在管理页面上，对权限所拥有者的权限进行灵活配置，同时也可以那个角色里有那些用户（帐号）等，都可以进行灵活的配置。

只能给角色设置权限，不能给用户组设置权限，本系统认为用户组只是一个临时的群组来处理，需要设置权限的群组按角色来处理。

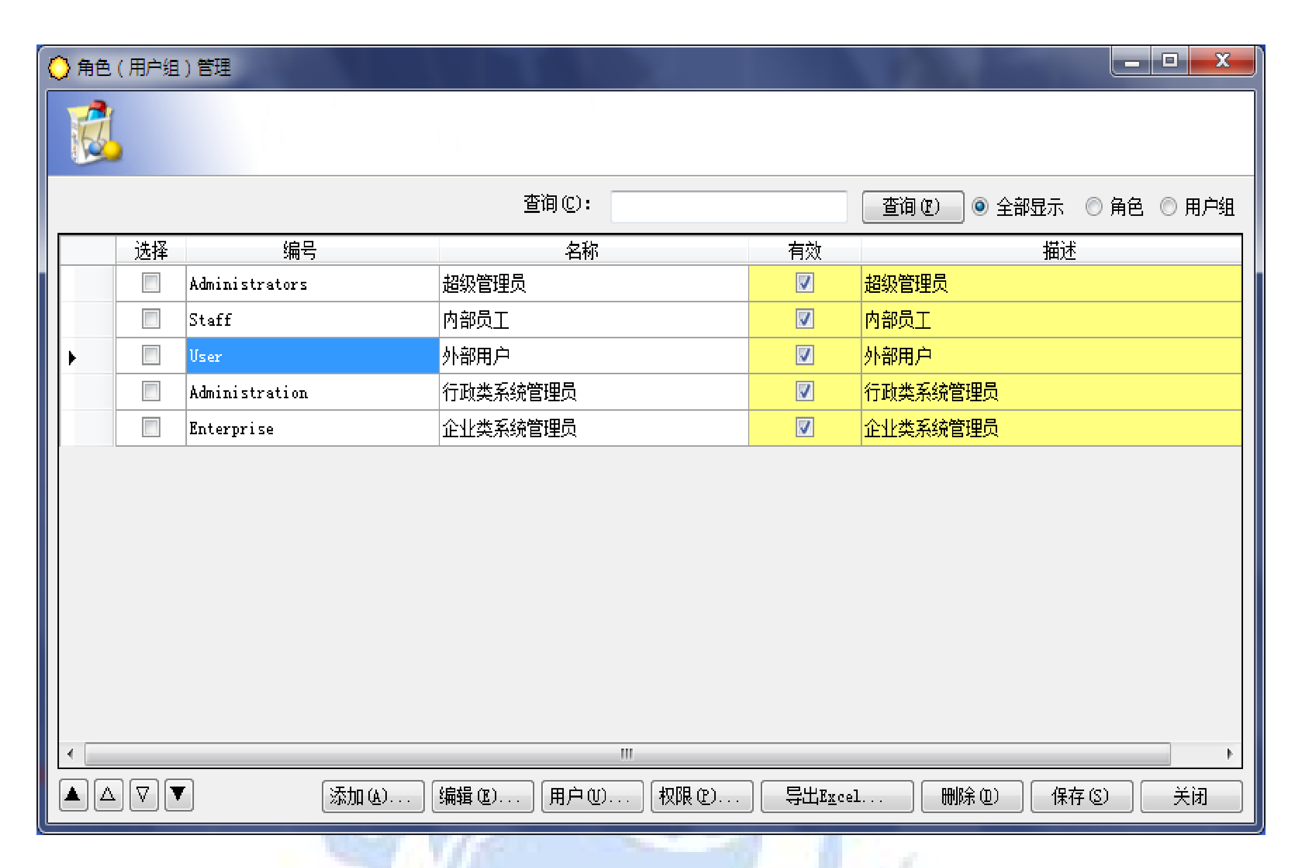


图 5-2 角色（用户组）管理界面

角色添加、编辑效果如下图，可以设置角色是否有效（停用）标志，不建议删除角色，添加、编辑时角色编号、名称都不允许重复，角色名称是必输项，按表头某个列近些排序后，需要点保存按钮后最新的排序顺序才会生效。

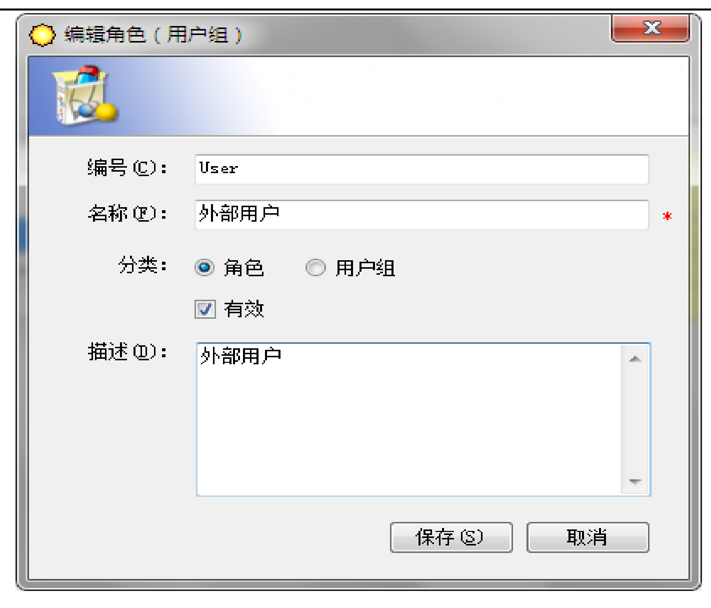


图 5-3 编辑角色（用户组）界面

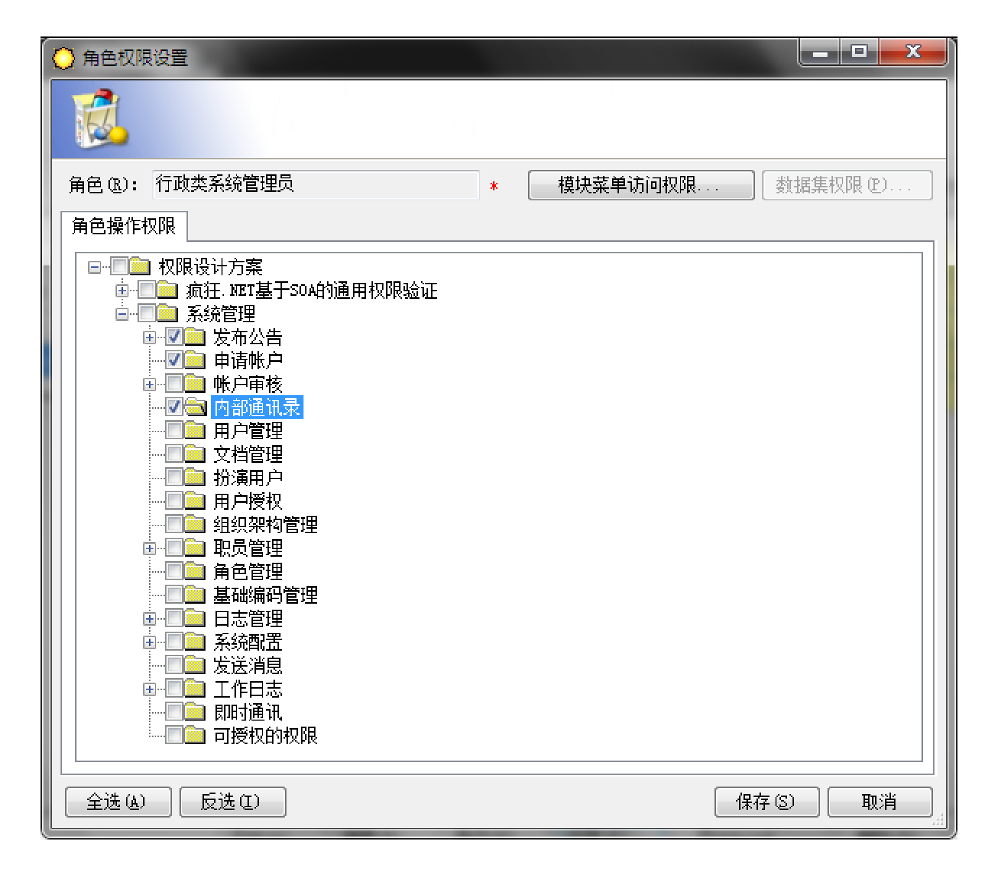


图 5-4 角色（用户组）权限设置界面

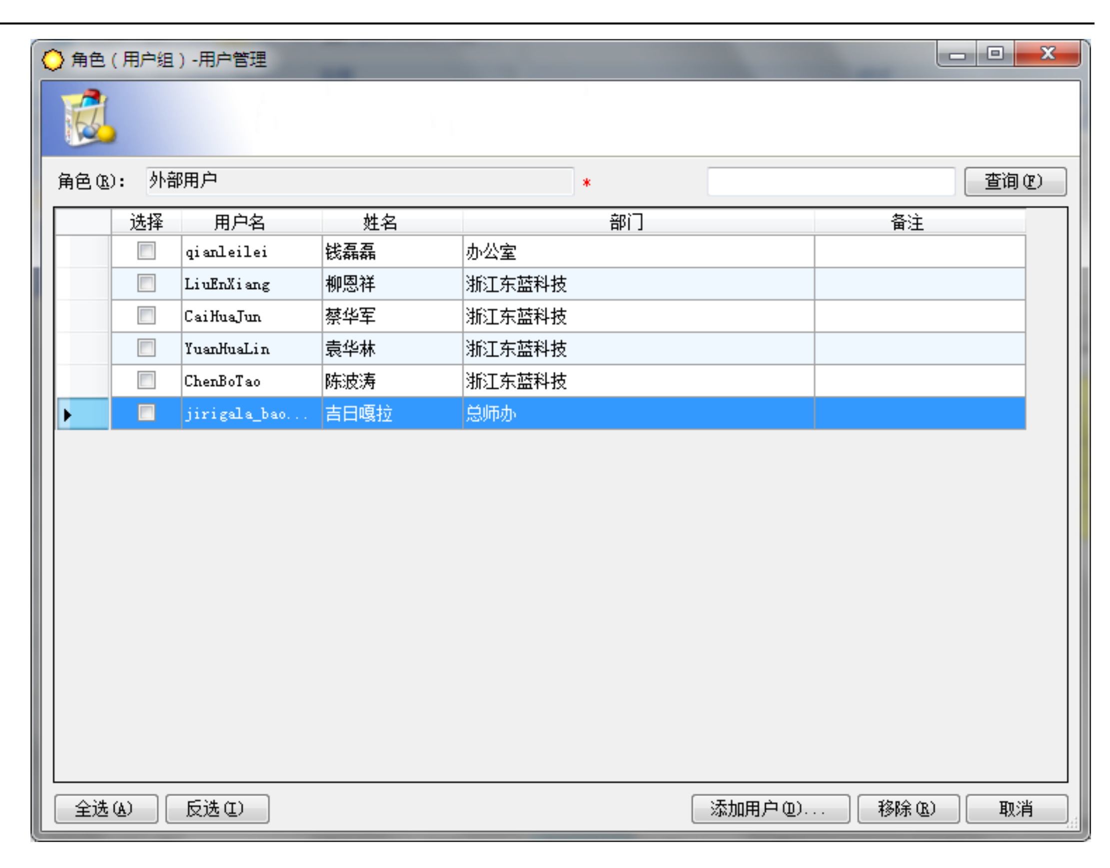


图 5-5 角色（用户组）- 用户（帐号）管理界面



5.2 角色的分类管理

当系统中管理的角色比较多时，管理起来会有些不方便、例如查找一个角色时就变得有些困难了。

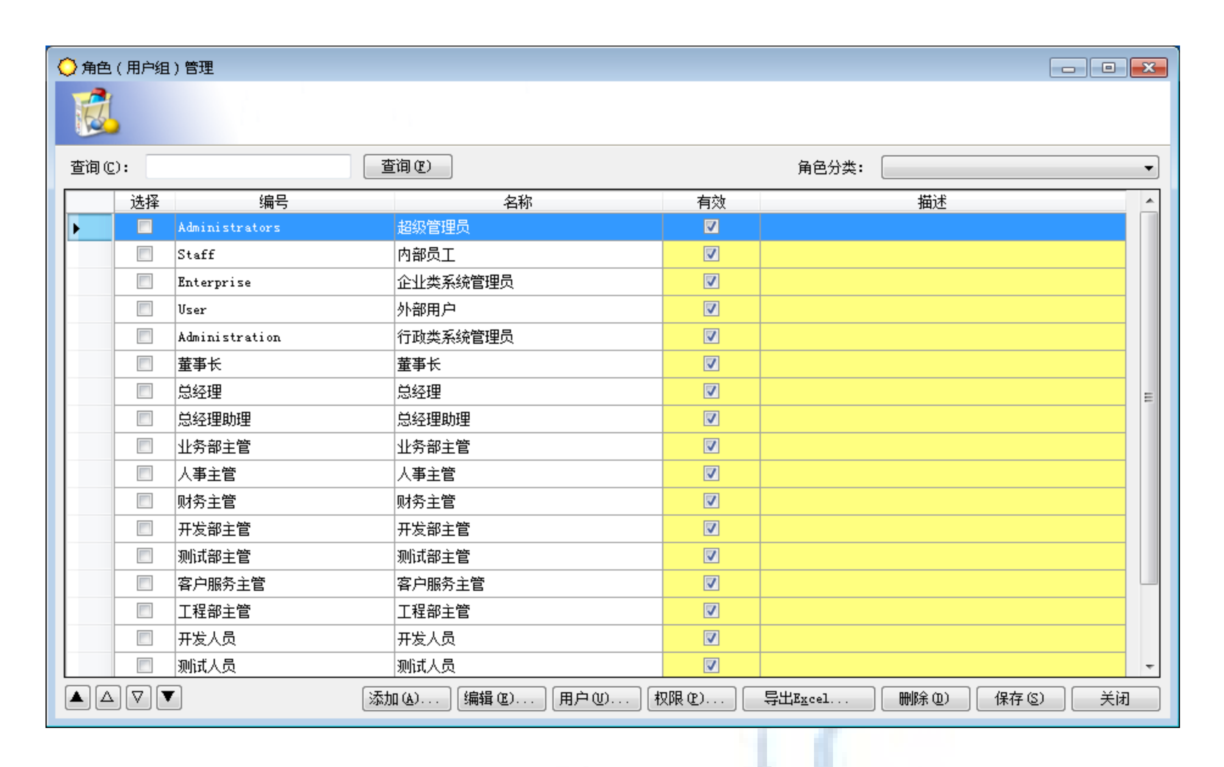


图 5-6 未使用分类管理用户列表界面

如果我们对角色进行适当的分类，那就会好很多，例如如下图效果，加入一个角色分类后，管理起来很方便了。其实系统角色、业务角色、职务职位都可以看作是一个角色的分类，他们都可以设置权限，设置哪些用户帐号归属于这个分类中。



图 5-7 使用分类管理用户列表界面

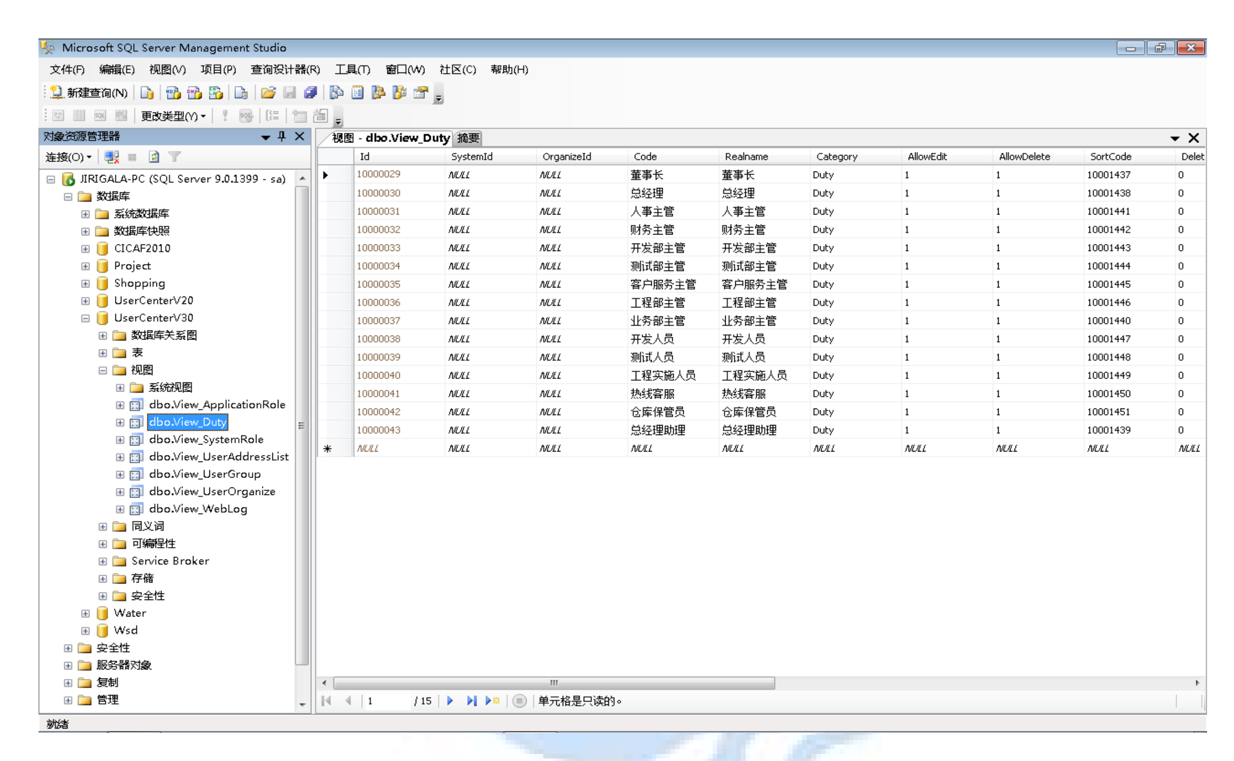


图 5-8 用户职位分类列表

我们再看看角色分类是如何管理的？说得俗一些，就是下拉框里的数据是怎么进行管理的？我们先点菜单“选项（下拉框）管理”。



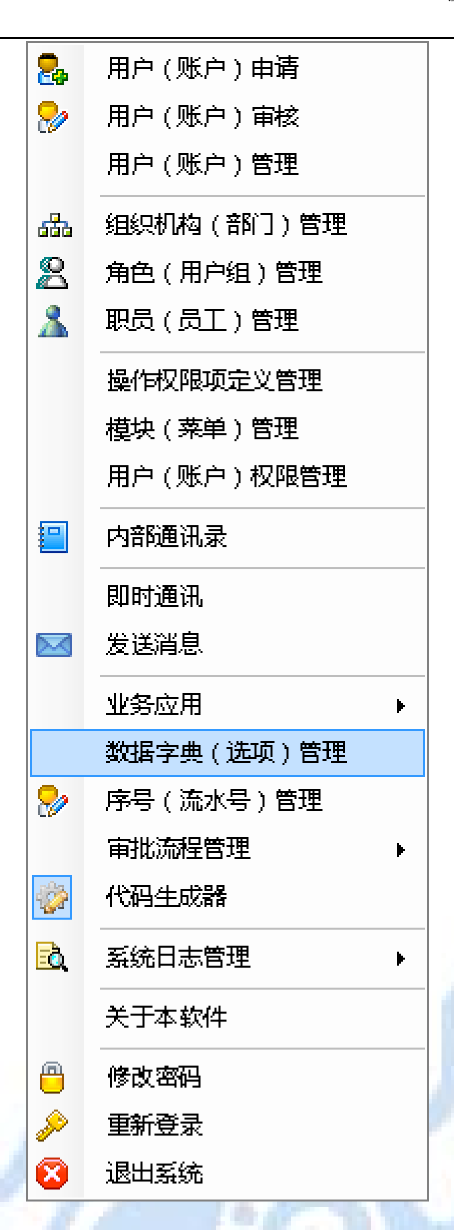


图 5-9 角色分类数据字典菜单

列表中有数据角色分类，然后点击 字典明细按钮，会弹出下面的数据窗口。



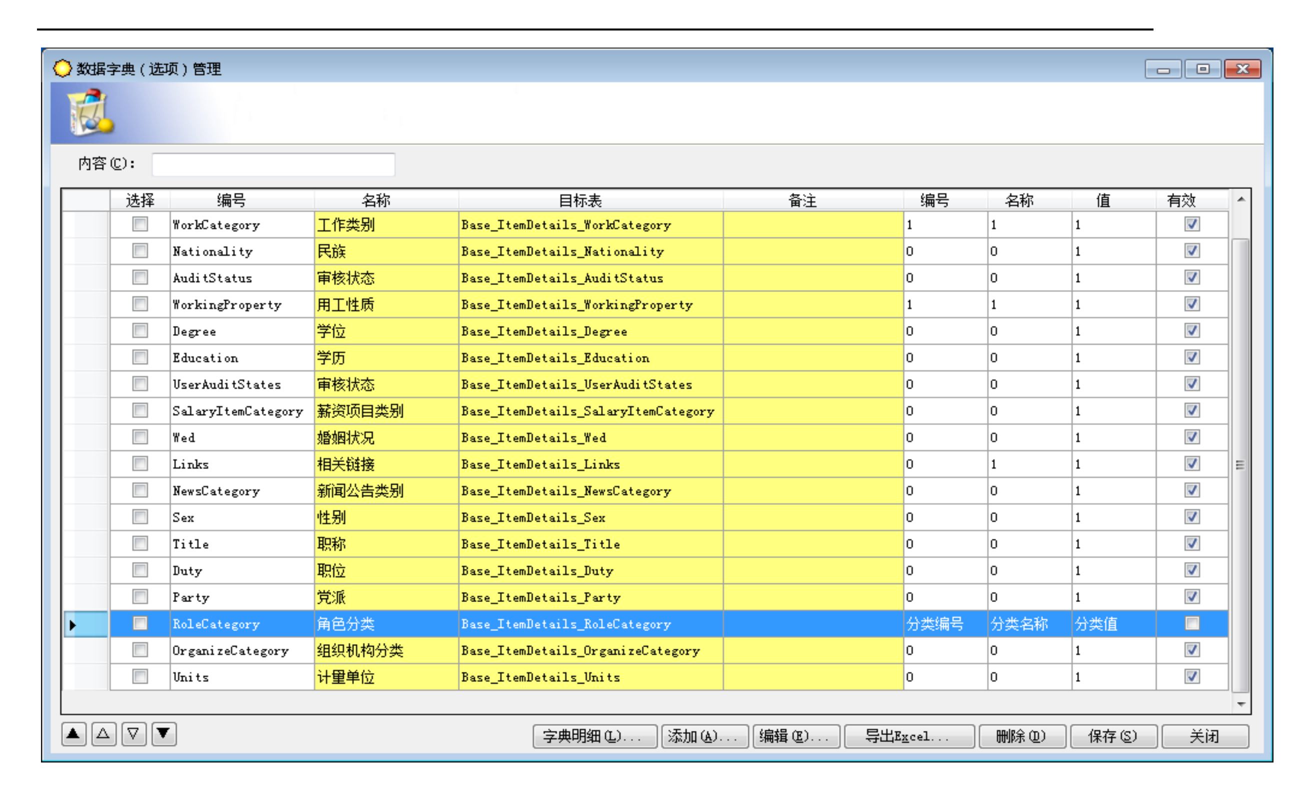


图 5-10 角色分类数据字典界面



图 5-11 角色分类管理界面



第6章 职员（员工）管理

职员管理功能模块主要是对公司内部员工数据进行管理，有新建员工数据、更改员工信息、员工部门变更等等。

职员（员工）管理功能主要是为了用在企业内部信息化管理系统开发的，在人力资源管理系统中的可二次开发。

6.1 职员（员工）管理

创建职员信息时可以同时创建用户帐号信息，可以将照片存在数据库中、当然此职员管理页面，可以按客户的个性化需要进行二次开发，可以将需要输入的内容都补充到这个功能界面上，这个界面只是一个快速的参考原型。

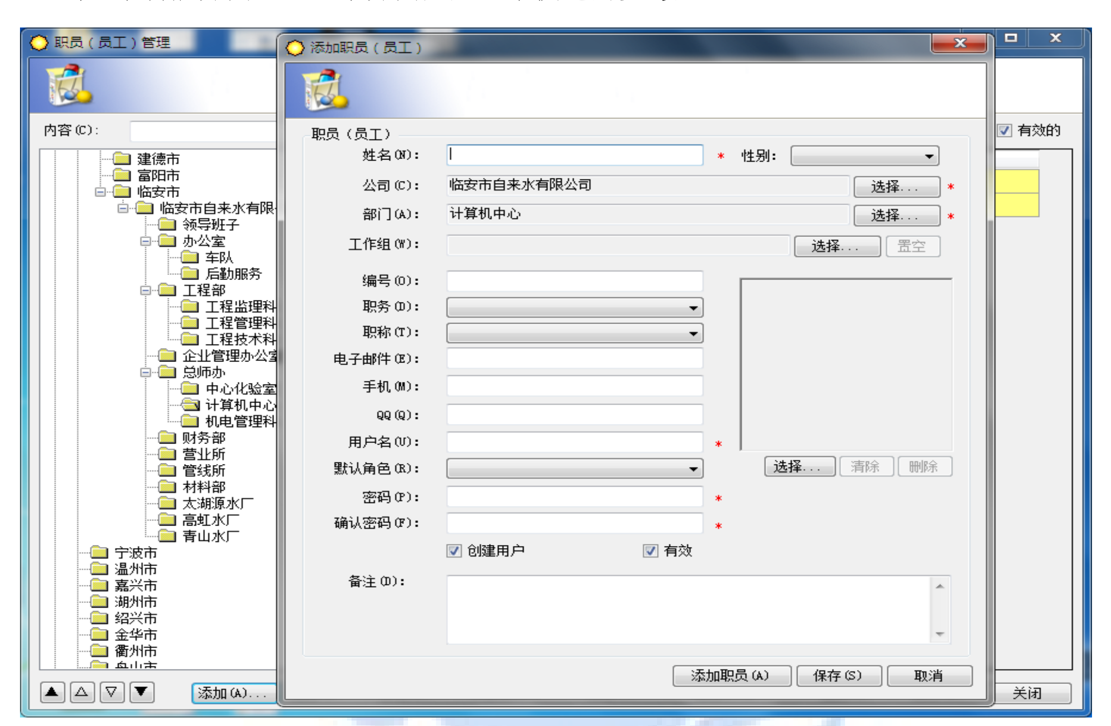


图 6-1-1 职员（员工）管理窗体

默认情况认为一个职员是属于某个组织机构，当然现实上有很多一个人兼职多个岗位、同时兼职多个组织机构的不同岗位的情况，本系统在数据库设计上，提供了一个职员可属于多个组织机构的扩展性，暂不提供支持此功能的相应的设置管理界面。

6.2 职员（员工）的排序顺序

技术人员往往不太注意细节，特别是程序员沉迷于程序代码之中，在日常工作中，特别是在中国，很在乎排名问题、例如公司的通讯录中的排名先后、往往有一些含义在里面，例如职位最高的排在最前面、地位最高的排在最前面，这些人的排序顺序随着职位的变化也需要及时的进行调整，日常工作中也常常也需要随时能灵活的导出这些数据，进行重复利用，快速生成一些报表数据等等用。

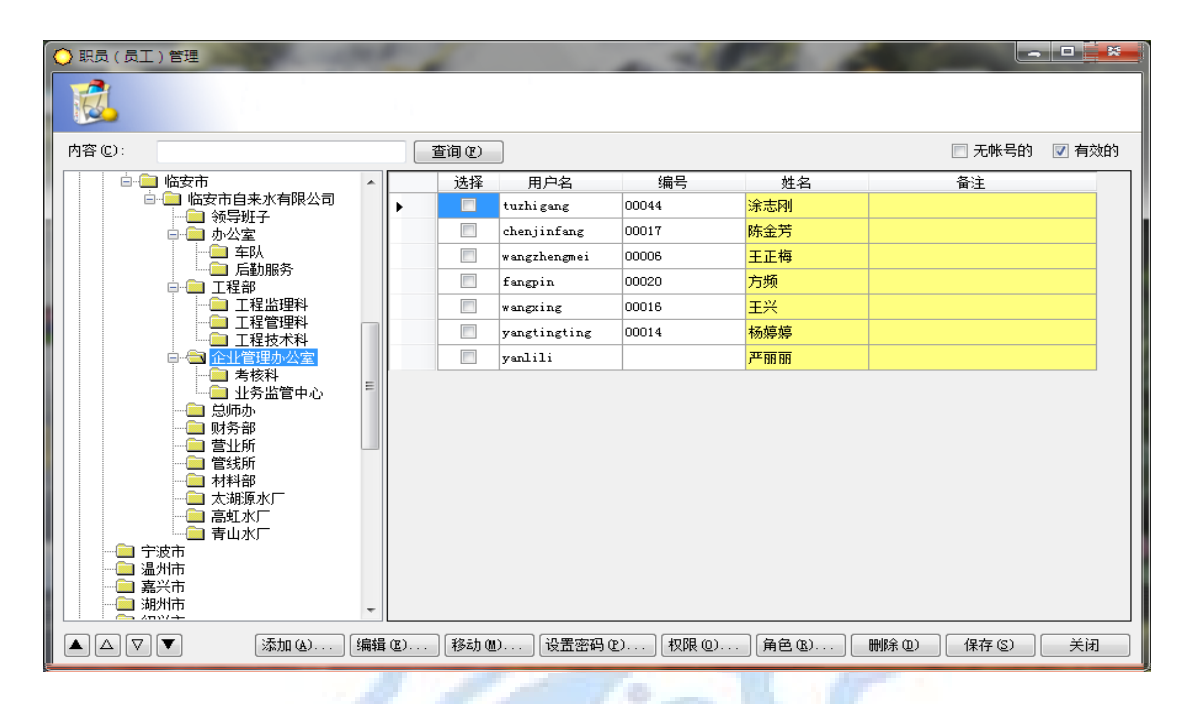


图 6-2-1 职员（员工）管理窗体

需要在应用软件里，不仅能灵活的设置员工的部门归属关系，而且还能快捷有效的设置部门中的排序顺序才能符合日常工作习惯、管理上的要求及客户的实际需求。

有些软件项目，就是因为人员的先后顺序不能灵活设定，导致整个软件用起来不顺手，或者找一个人变成很困难的情况，人员的排序顺序乱了，寻找一个人也会变得很麻烦，不能适应日常习惯，友善度会大大降低。

排序码需要人工输入操作上也有些不友好，相对智能的比较好些，不用每个人一个个都调整排序码，只是点点选选就可以了，客户会越来越喜欢用的这样智能的软件。

6.3 职员（员工）与用户（帐号）的关系

不是所有的职员（员工）都有用户（帐号），例如公司的保洁员，可能是这个公司的正式职员，但是未必有登录公司业务系统的用户（帐号），有用户帐号的未必全是公司职员，例如公司的客户、供应商也有登录公司业务系统的用户帐号，但是这些用户帐号不是公司的职员。

有的系统需要有内部职员管理模块例如大多数内部业务信息管理系统，有的系统是外部网站类新闻类系统性质的、这类的系统并不关注内部职员的管理模块。

作为一个工具类、可进行二次开发的软件组件，在设计上遵守了最简洁数据结构。所以将内部职员的信息与用户帐号的信息进行了隔离设计，这样系统可以以最简洁的用户帐号结构与其他系统可以进行集成工作，同时又能支持丰富的职员信息管理的内部信息化项目的需要。

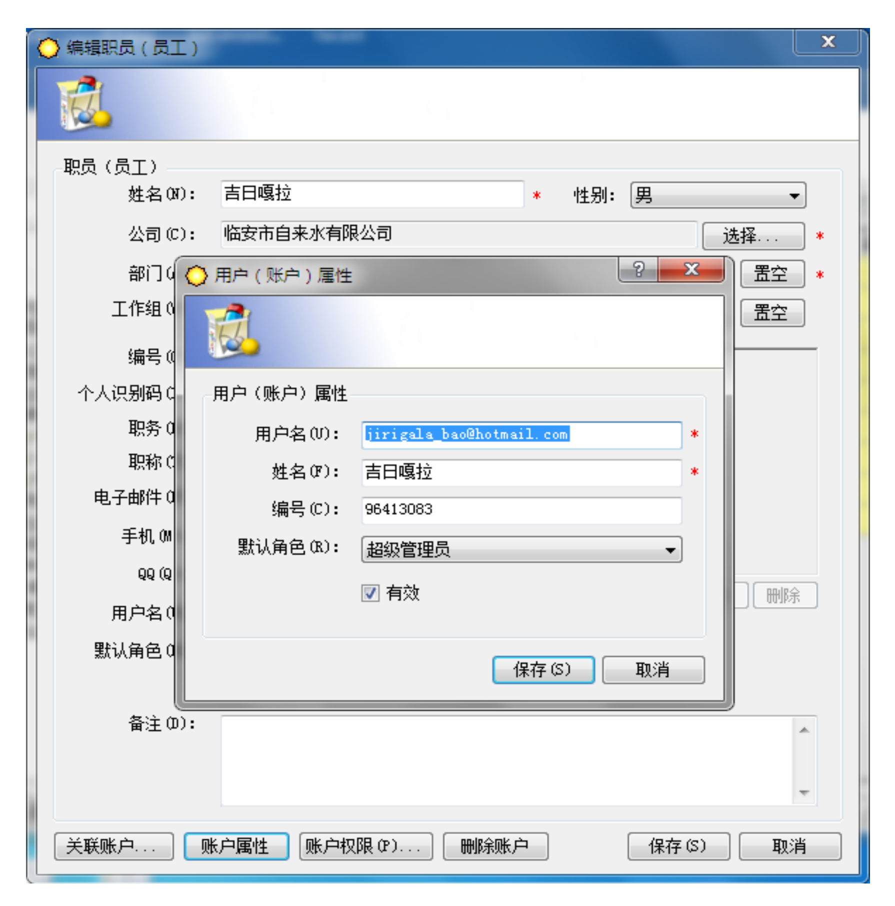


图 6-3-1 用户（帐号）属性

6.4 职员（员工）导出数据

往往系统只能提供的是有限的数据查询统计功能，甚至报表都是固定的，无法按客户的个性化要求统计查询数据，并打印出相应的统计数据，这时建议有数据的导出功能，可以将数据都导出到 Microsoft Office Excel 文件中，然后客户按自己的要求，进行随意的统计过滤、按个性化的要求进行打印输出相应的统计报表。

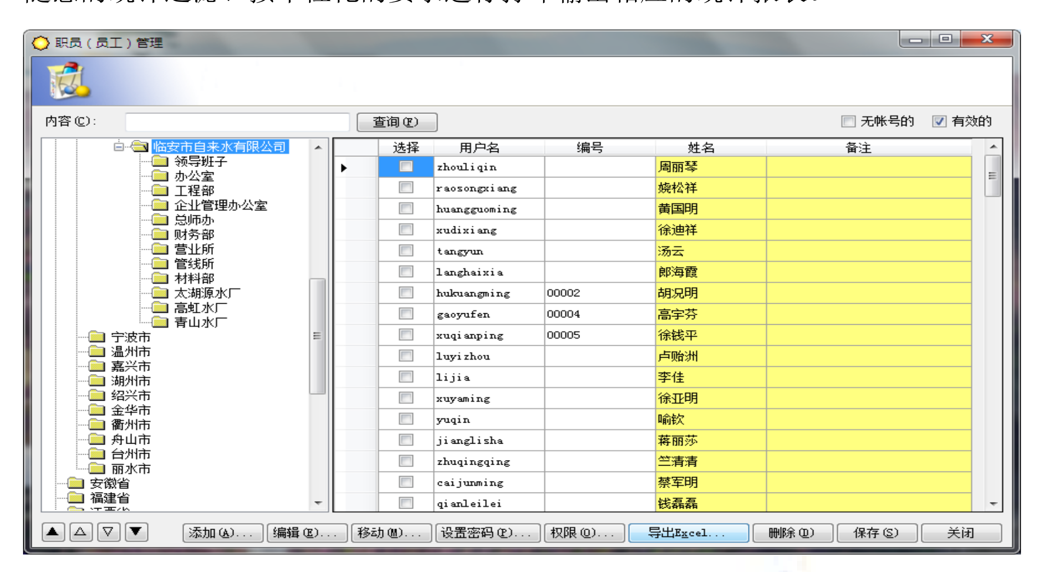
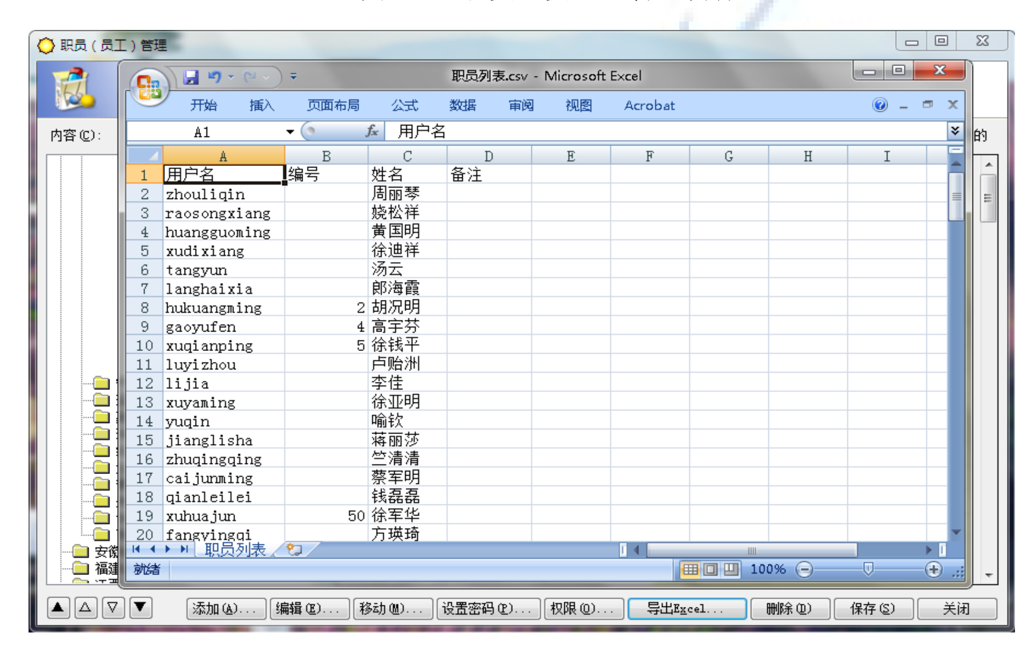


图 6-4-1 职员（员工）管理窗体



	A	B	C	D
1	用户名	编号	姓名	备注
2	zhouliqin		周丽琴	
3	raosongxiang		娆松祥	
4	huangguoming		黄国明	
5	xudixiang		徐迪祥	
6	tangyun		汤云	
7	langhaixia		郎海霞	
8	hukuangming	2	胡况明	
9	gaoyufen	4	高宇芬	
10	xuqianping	5	徐钱平	
11	luyizhou		卢贻洲	
12	lijia		李佳	
13	xuyaming		徐亚明	
14	yuqin		喻钦	
15	jianglisha		蒋丽莎	
16	zhuqingqing		竺清清	
17	caijunming		蔡军明	
18	qianleilei		钱磊磊	
19	xuhuajun	50	徐军华	
20	fangyingqi		方瑛琦	

图 6-4-2 导出数据效果

6.5 职员（员工）离职处理

一个公司，特别是软件开发类公司里，由于到处充满了机会，经常会有新员工入职、老员工离职、岗位变动等等事情发生得也会频繁，员工离职时，不应该把员工的相关数据删除掉，而是把员工账号的有效标识设置为无效状态才对。



图 6-5-1 设置职员（员工）的属性

将界面上的，有效标注选中状态去掉，就表示此员工的账号被设置为无效状态了，这个用户就不可以登录公司的相关信息系统了，将有效标志选中状态恢复过来后，此用户就允许登录到相应的信息系统中。

当然一个职员的离职相关手续应该是在人事系统里进行配置管理的，可以在客户的人事系统里，通过调用二次开发接口、直接用程序设置为无效状态，这样没必要系统管理员进行重复性的设置工作了。



第7章 内部通讯录

经常会遇到找不到某个同事的联系方式或是某个新来同事的联系方式怎么也找不到，公司发放的通讯录里还有一些很早已离职人员的联系方式等。公司需要及时整理最新的内部通讯录，高效沟通协调，每次都人工更新通讯录一方面是工作效率低，另一方面这个工作也很繁琐、没有技术含量，重复性也多，经常有变动。

当进入一个公司后，往往不知道哪个是哪个，谁叫什么名字，谁是负责哪个方面的？若有一个内部通讯录，可以比较轻易的看到哪个人的照片是什么样的？这个人又叫什么名字，这个人又是负责哪个方面的，那可以用更短的时间了解公司的人员结构，更快融入到公司工作当中。

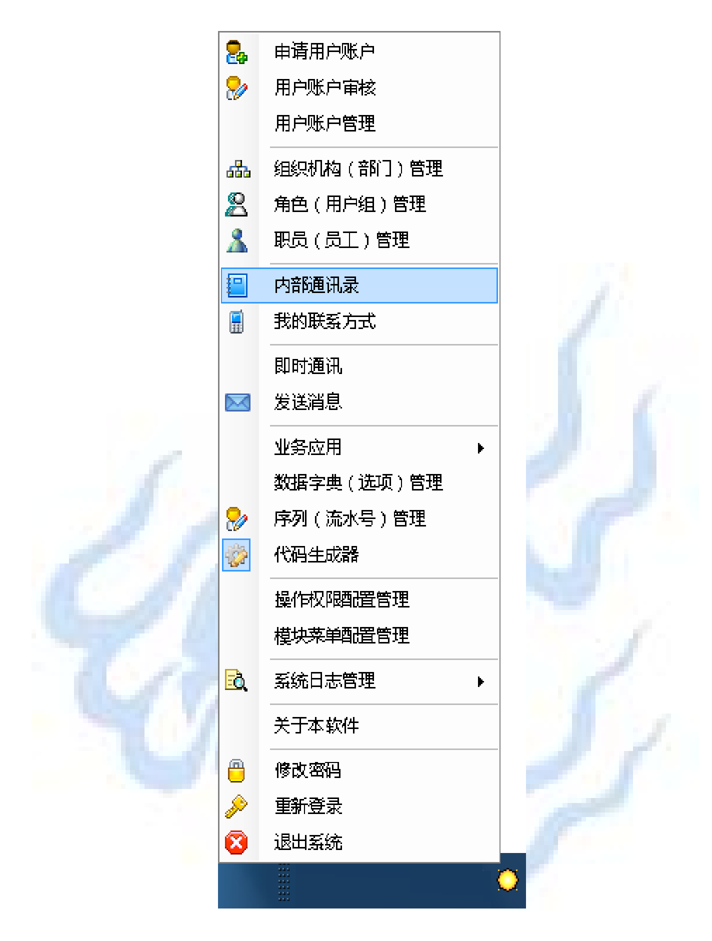


图 7-1 内部通讯录菜单



7.1 我的联系方式

当前的用户若是公司的内部职员（员工），登录系统后会出现“我的联系方式”菜单，点击此菜单可以设置自己的最新联系方式。



图 7-1-1 编辑通讯录窗体

可以在本窗体中还可以设置自己的照片，支持拖动照片功能，直接把照片文件拖入照片控件上就可以了。

7.2 内部通讯录



图 7-2-1 内部通讯录窗体

在内部通讯录窗体里，可以设置新员工的通讯方式、也可以快速设置自己的联系



方式，可以将当前的通讯录导出到 Excel 文件中。

内部通讯录可以按公司/部门快速过滤数据，方便差找、导出某个分公司的通讯录或者某个部门的通讯录。

可以通过邮件方式将最新的内部通讯录发送给所有的内部员工。

有操作权限的人员能编辑任何一个用户的联系方式，当然也可以批量快速编辑公司的通讯方式。

当然，只有相应的管理操作权限的操作者才可以更改内部员工的联系方式，没有操作权限的用户只能查看内部通讯录、只能修改自己的联系方式。





第8章 即时通讯

即时通讯功能主要是为了实现管理系统的通知信息能即时发送给相应的用户（帐号），为了实现即时高效的通知功能。虽然与商业化的即时通讯软件无法匹敌，但是往往内部个性化的管理系统，还是需要即时通知、即时沟通。

8.1 发送消息

可以通过发送消息功能菜单，给系统的中的用户（帐号）发送即时消息，若对应的用户在线状态或者登录了管理系统的页面，都应该能及时显示出来站内的即时消息。

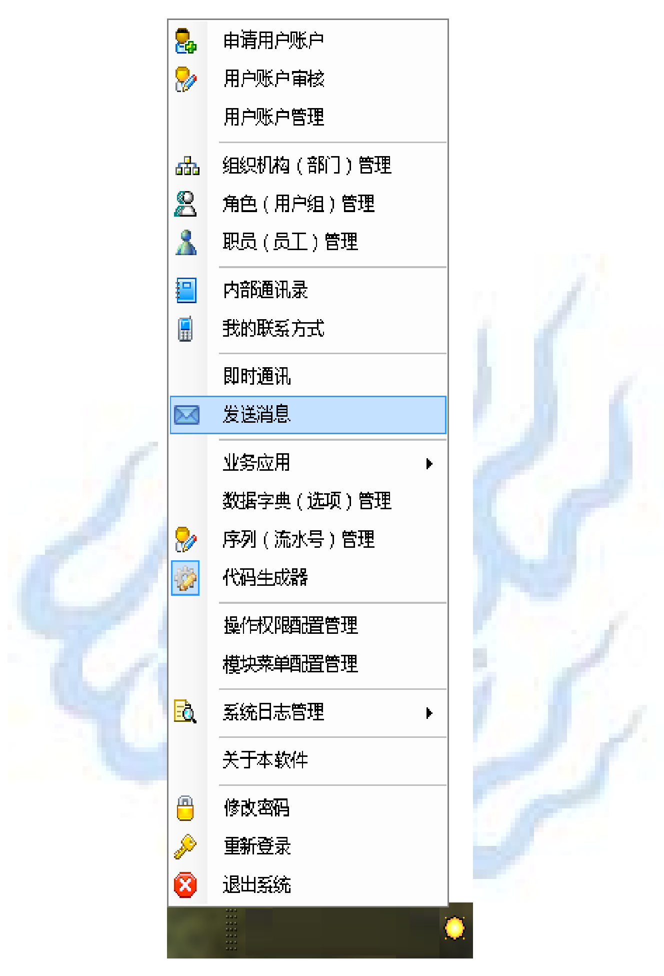


图 8-1-1 发送信息菜单

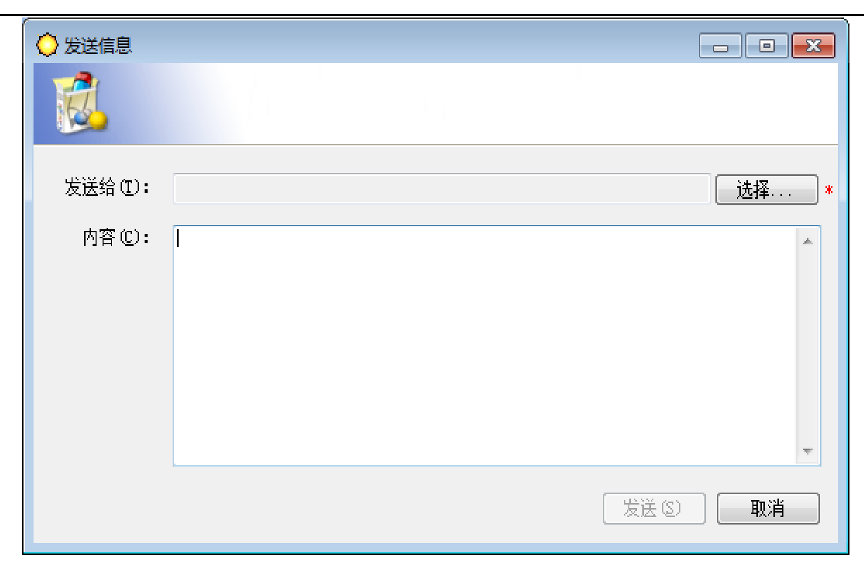
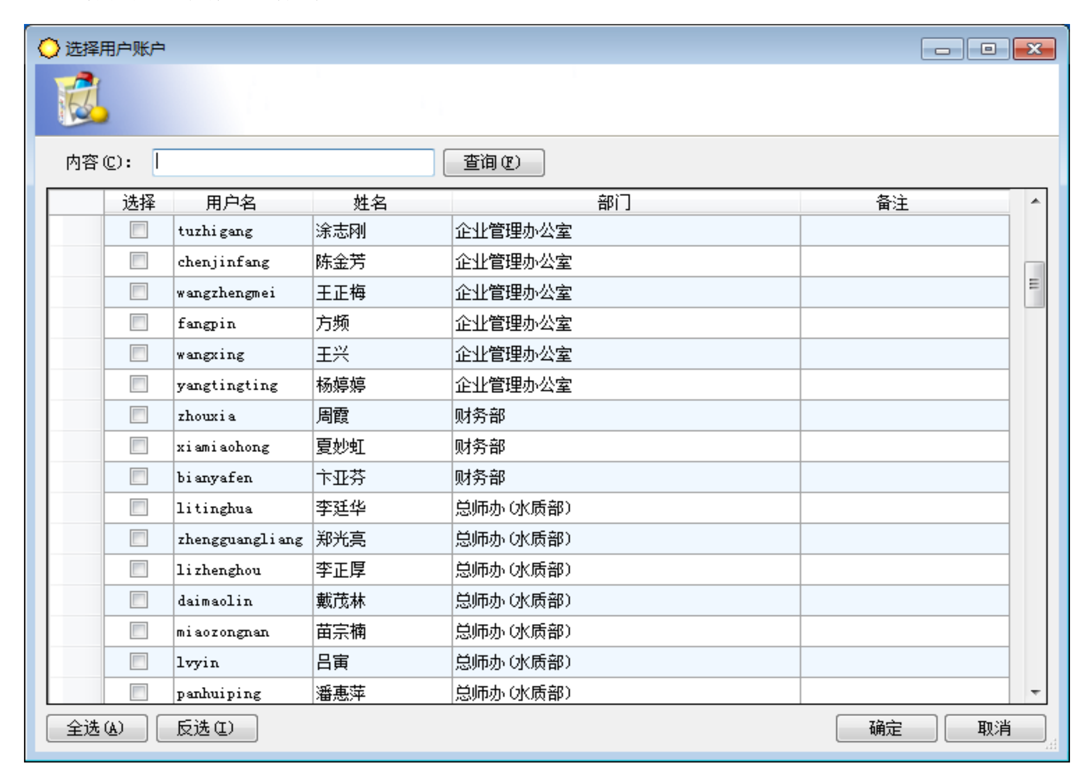


图 8-1-2 发送信息窗口

进入发送信息窗口后，可以选择发送的对象，按选择按钮弹出选择用户（帐号）的窗口，选择好目标用户（帐号）。输入内容后点发送（S）按键后，即时信息就可以发送给相应的用户（帐号）。



选择	用户名	姓名	部门	备注
	tuzhigang	涂志刚	企业管理办公室	
	chenjinfang	陈金芳	企业管理办公室	
	wangzhengmei	王正梅	企业管理办公室	
	fangpin	方频	企业管理办公室	
	wangxing	王兴	企业管理办公室	
	yangtingting	杨婷婷	企业管理办公室	
	zhouxia	周霞	财务部	
	xiamiaohong	夏妙虹	财务部	
	bianyafen	卞亚芬	财务部	
	litinghua	李廷华	总师办(水质部)	
	zhengguangliang	郑光亮	总师办(水质部)	
	lizhenghou	李正厚	总师办(水质部)	
	daimaolin	戴茂林	总师办(水质部)	
	miaozongnan	苗宗楠	总师办(水质部)	
	lvyin	吕寅	总师办(水质部)	
	panhuiping	潘惠萍	总师办(水质部)	

图 8-1-3 选择用户（帐号）窗口

8.2 即时通讯

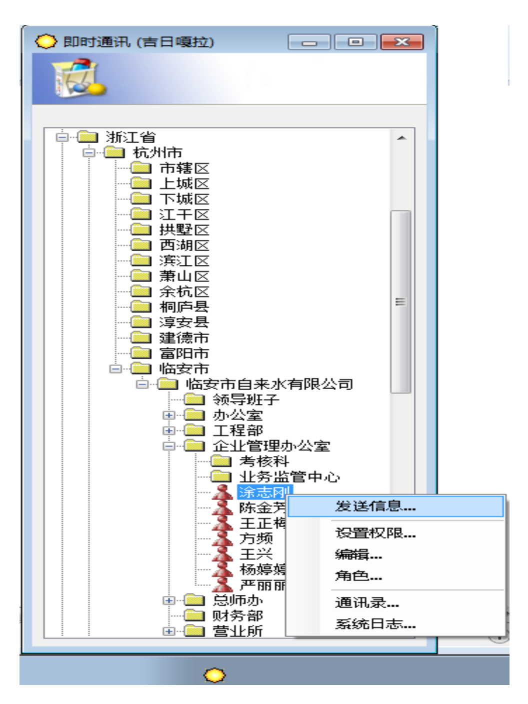


图 8-2-1 即时通讯

可以通过双击系统图标，出现当前的系统中的内部组织架构图，可以直接选择组织机构中的用户（帐号）直接发送消息，设置用户的帐号的权限等。

若有在线消息出来，可以直接弹出消息，也可以直接回复消息，用户的在线状态会时时更新，与 QQ、MSN 的在线状态很类似。

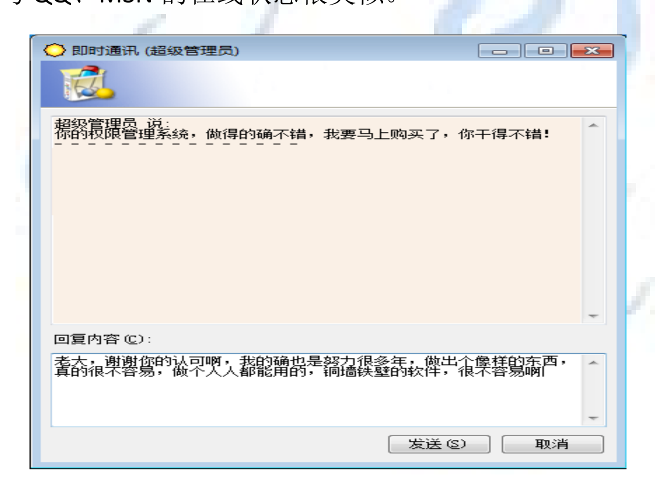


图 8-2-2 即时信息回复

上图为回复及时信息的功能效果图。



第9章 选项（下拉框）管理

数据字段（选项）管理模块主要是为了灵活控制整个系统中的可选项数据，可以按用户的习惯或者用户的个性化要求进行灵活管理。

同时这部分数据也可以在开发多种业务系统时有一定的重复利用价值，将数据导入到新的业务系统里重复利用，一些选项数据就不用重新反复录入了，可以提高新项目的开发实施速度。



图 9-1 选项（下拉框）管理菜单

9.1 选项（下拉框）管理

这些各种可选项数据经过多年的维护后，也可以当是一种知识库、随着公司的业务积累，越来越完善、越来越齐全，合理性也会更强一些。

某些业务管理系统，就是由于满足不了最终用户的一点点个性化的需求，在日常办公中不能适应客户的实际需求，导致客户使用不便、达不到客户对业务管理系统的人性化方面的要求，最终由于一些小小的需求满足不了客户的要求，导致整个项目实施不

顺。

若在开发阶段进行了充分的考虑、多站在不同客户的个性化需求角度进行了设计，那开发出来的系统会更符合客户的实际工作需要，客户也更容易接纳设计出的软件管理系统。

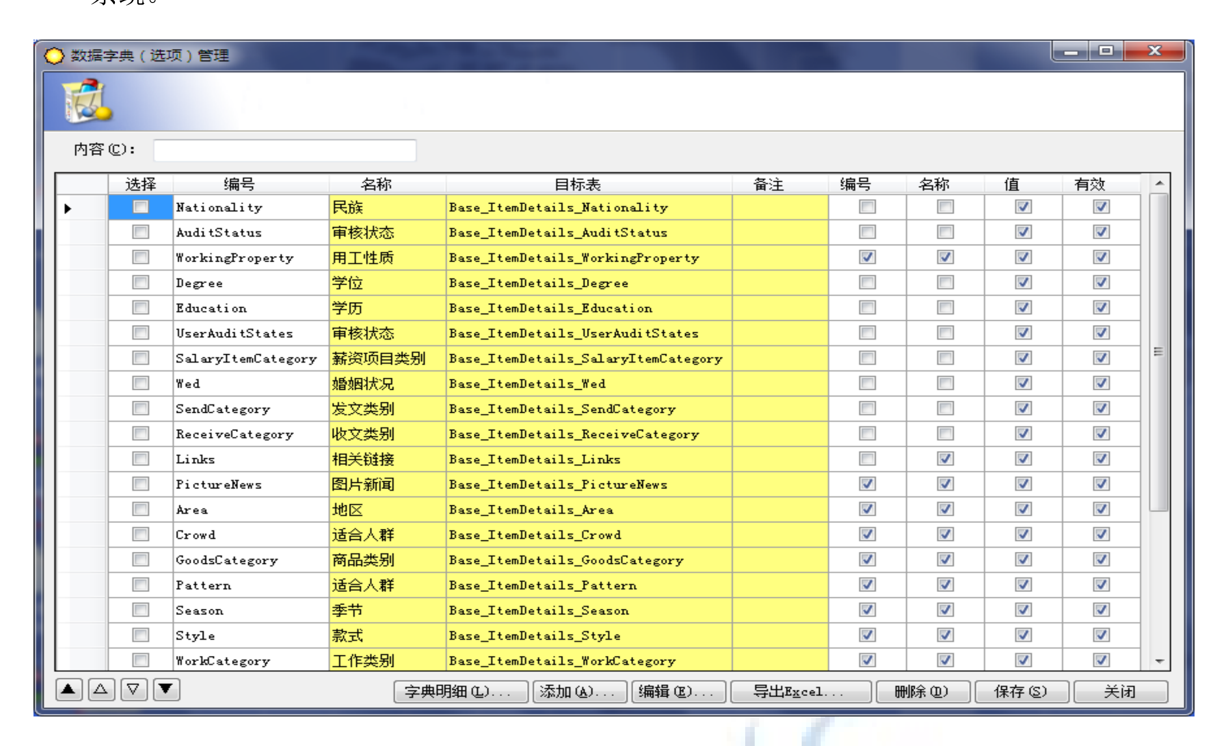


图 9-1-1 选项（下拉框）管理窗体

上图中的每一条数据项对应了后台的一个数据表，把每个选项详细情况放在原始数据结构统一目标表里。这样设计的好处在于：一方面可以任意扩展目标表，增加字段，对其他的目标表没有影响，可以按开发的需要，增加任意字段进行扩展；另一方面，导入导出数据更加灵活，可以将部分积累好的选项数据快速导入到另外一个信息管理系统中，可以明显提高工作效率，提高数据的重复利用程度，这些数据也是公司的重要业务知识积累部分。

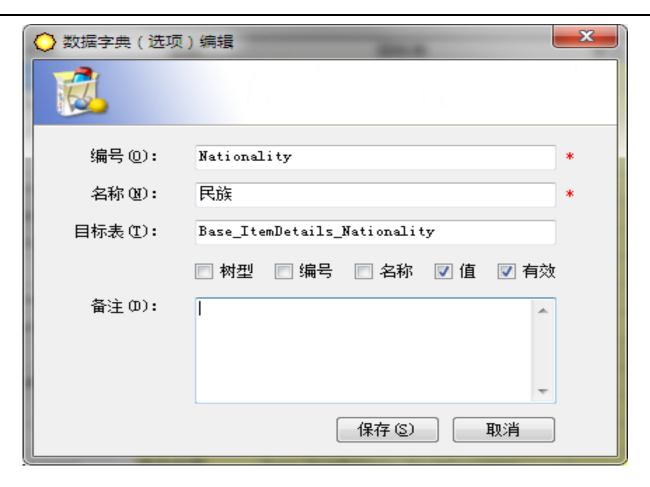
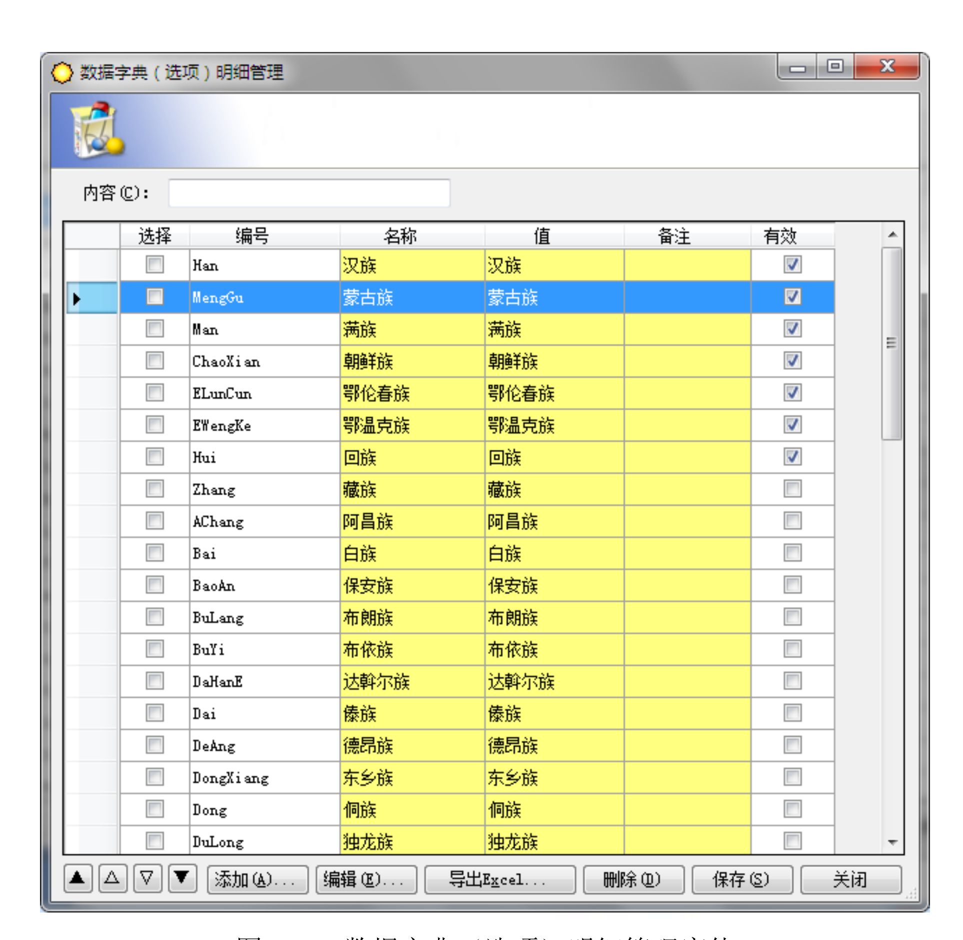


图 9-1-2 数据字典（选项）编辑窗体

9.2 数据字典（选项）明细管理



选择	编号	名称	值	备注	有效
☐	Han	汉族	汉族		☑
☐	MengGu	蒙古族	蒙古族		☑
☐	Man	满族	满族		☑
☐	ChaoXian	朝鲜族	朝鲜族		☑
☐	ELunCun	鄂伦春族	鄂伦春族		☑
☐	EWengKe	鄂温克族	鄂温克族		☑
☐	Hui	回族	回族		☑
☐	Zhang	藏族	藏族		☐
☐	AChang	阿昌族	阿昌族		☐
☐	Bai	白族	白族		☐
☐	BaoAn	保安族	保安族		☐
☐	BuLang	布朗族	布朗族		☐
☐	BuYi	布依族	布依族		☐
☐	DaHanE	达斡尔族	达斡尔族		☐
☐	Dai	傣族	傣族		☐
☐	DeAng	德昂族	德昂族		☐
☐	DongXiang	东乡族	东乡族		☐
☐	Dong	侗族	侗族		☐
☐	DuLong	独龙族	独龙族		☐

图 9-2-1 数据字典（选项）明细管理窗体

例如一个人力资源管理系统，可能在同一个软件，在南方实施与北方实施，客户的要求都会有些不同，在大西南实施与东北实施的客户的要求差别也会很大，大西南实



施时需要有很多民族选项，很多民族可以从列表里能选择比较适应客户的需要，若在浙江杭州临安实施一个人力资源软件几乎只需要列出2-3个民族就可以了，很少有少数民族，所以使用时也不需要把其他民族都列出来，能设置其他民族选项为无效状态比较好，也不是说删除才对，应该是有需要时，再把这些设置为有效就可以了。

当然不只是设置为有效无效状态就可以了，还能设置排序顺序比较好，例如在内蒙古实施这个人力资源软件，应该蒙古族排序在第2个位置比较好，因为在内蒙古也是汉族人数最多蒙古族人数是其次，当然若在一个民族中学实施这个人力资源管理系统，那应该把对应的少数民族放在第一个位置，更适合用户的使用习惯。

所以一些选项的管理的排序顺序、是否有效状态、是否可以灵活增加选项是非常有必要的，这样软件的多种类型用户的适应性会更强一些。





第10章 系统日志管理

系统里的任何操作以及任何的模块的访问和任何服务的调用，用户的登录情况等等都会被记录起来，方便日后追查之用。

若记录日志影响了系统的整体性能，不需要进行日志记录功能，可以在配置文件中设置开关选项，可以取消日志记录功能。

日常应用中，不建议去掉日志记录功能，日志管理模块主要由用户（帐号）访问情况、按用户（帐号）查询，按模块（菜单）查询，按日期查询等四个功能部分组成。

系统日志管理模块也有彻底清理正式上线前的日志的功能，方便实施人员使用、方便优化系统性能、减少数据库容量、减轻数据库负担，同时定期维护、定期整理系统也是很有必要的后期维护工作内容之一。

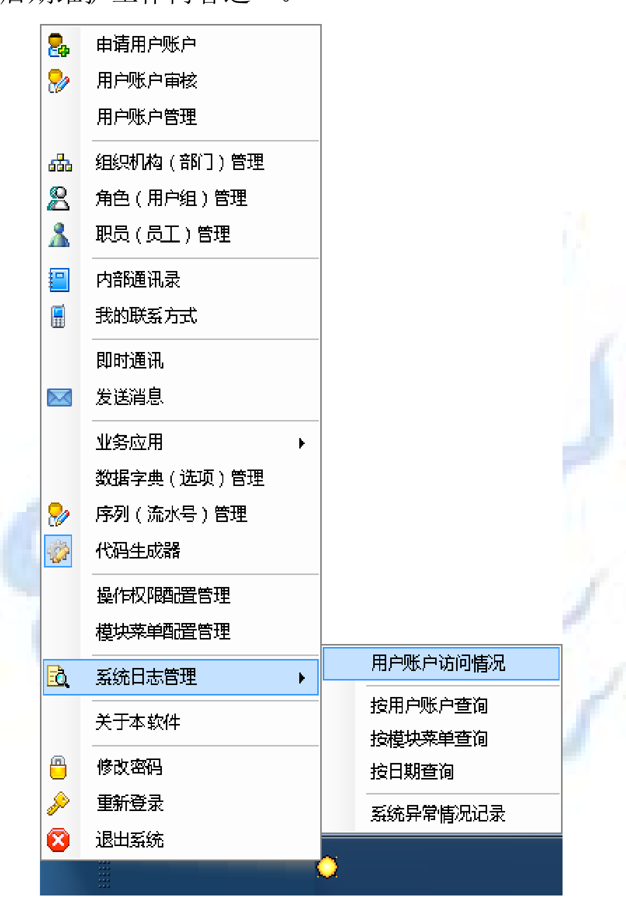


图 10-1 系统日志管理菜单

10.1 用户（帐号）访问情况

访问情况管理主要对系统的访问情况进行管理，该模块记录了所有用户登录系统情况、最早访问时间、最后访问时间、访问次数、访问 IP 地址等等，并可以显示当前在线的用户情况，同时可以将访问情况导出到 Excel 文件中，在系统正式上线前，还可以进行重置访问情况的功能，当然还可以按各种字段进行排序等。



图 10-1-1 用户（帐号）访问情况窗体

有些用户系统都没登录过，就会说这个系统不好用，有些功能模块都没用过几次就说这个系统不好等等都养成了习惯了，这些问题以通过检查用户的访问情况来体现出来真实情况，这个用户是否真的用了系统？是否在经常用？到底用了几次？一目了然了。

10.2 按用户（帐号）查询

按用户（帐号）查询功能主要是可以按某个用户（帐号）的访问情况、使用情况，按指定的用户（帐号）的问日志记录进行过滤，同时可以导出查询后的结果进行二次处理。

默认显示当天的访问情况记录，开始日期、结束日期可以自由设置任意查询，在本窗口中可以清除全部的日志记录，只有超级管理员有此权限，其他用户不可以清除日志记录。

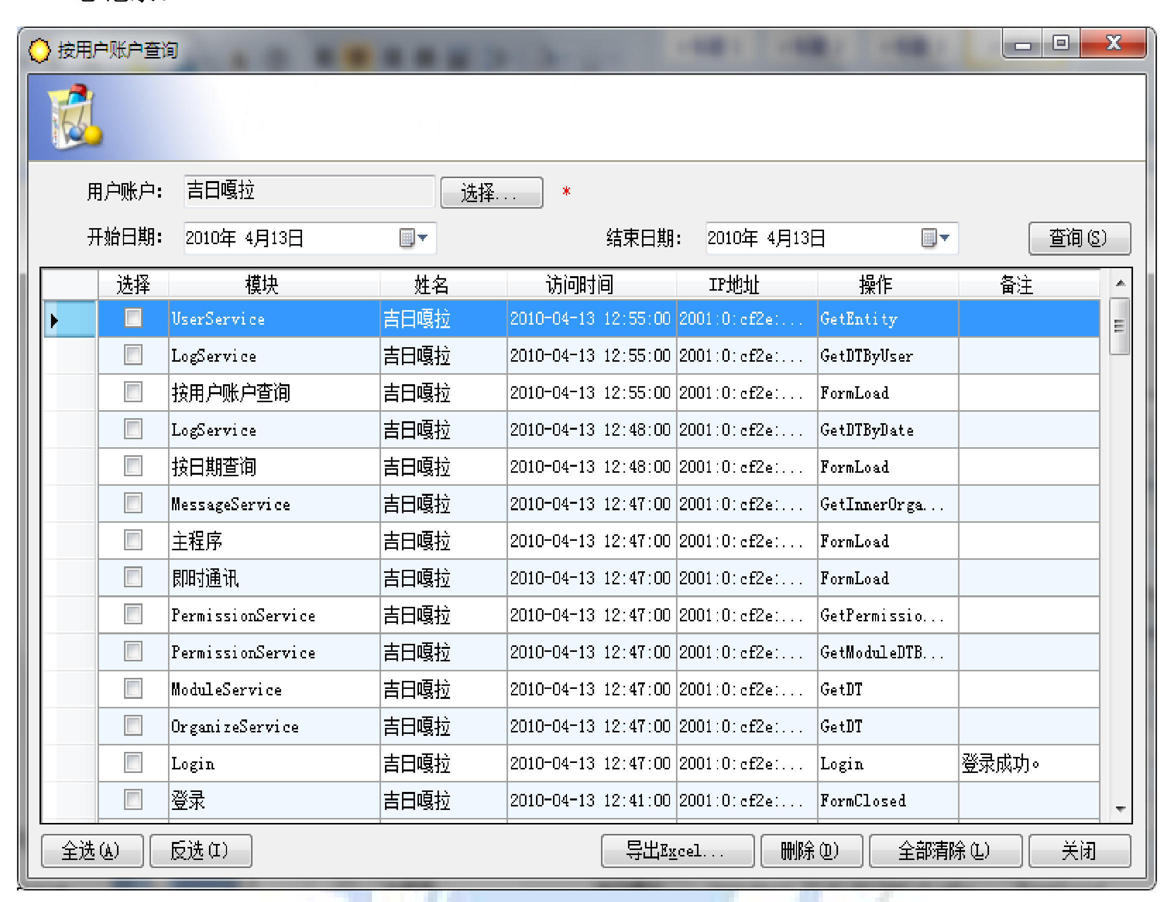


图 10-2-1 系统日志按用户（帐号）查询窗体

10.3 按模块（菜单）查询

按模块（菜单）查询功能主要是可以按某个功能模块的访问情况、使用情况，按指定的菜单的访问日志记录进行过滤，同时可以导出查询后的结果。

默认显示当天的访问情况记录，开始日期和结束日期可以自由设置，任意查询都

可以在本窗口中可以清除全部的日志记录，只有超级管理员有此权限，其他用户不可以清除日志记录。

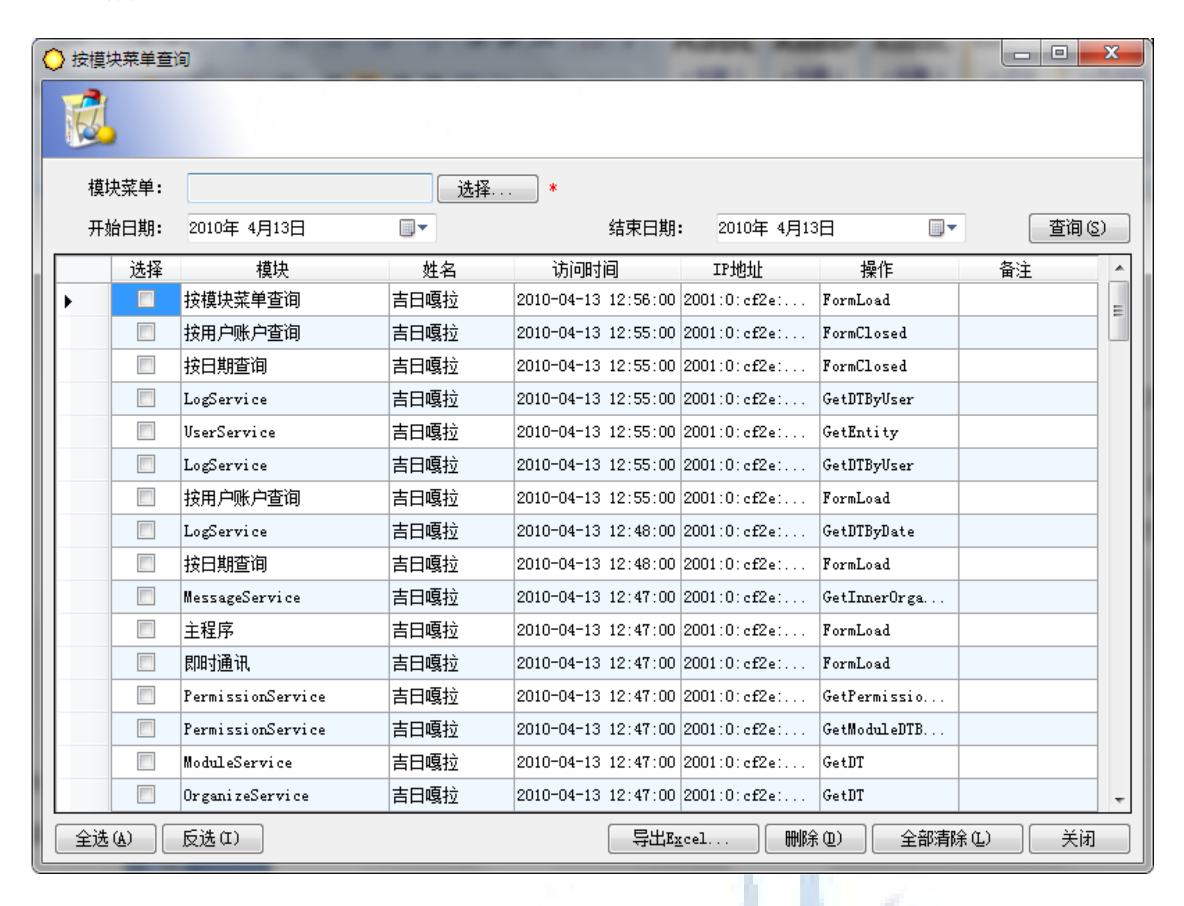


图 10-3-1 系统日志按模块（菜单）查询窗体

10.4 按日期查询

按日期查询功能主要是可以按某个日期范围差找系统的访问情况和使用情况，按日期范围排查系统的日志记录，可以导出查询后的结果进行二次处理。

默认显示当天的访问情况记录，开始日期、结束日期可以自由设置，任意查询，在本窗口中可以清除全部的日志记录，只有超级管理员有此权限，其他用户不可以清除日志记录。



图 10-4-1 按日期查询窗体



第11章 模块（菜单）管理

模块（菜单）管理模块是针对开发人员和后台管理员对系统进行管理为目的，此模块并不是针对普通的用户日常使用，在日常管理维护过程中不建议对此功能模块有任何修改，已确保系统的正常运行，尽量避免没必要的麻烦产生比为好。

模块（菜单）的配置通常是在数据库表中保存，功能模块繁多时直接操作数据库难免会产生眼花缭乱的感觉，有一个快捷的管理界面对开发人员来讲还是很有必要的。

模块（菜单）管理是对系统的功能模块（菜单）进行整理维护和设置功能模块的有效性（停用标志）、菜单名称、排序位置、操作权限关联等等。

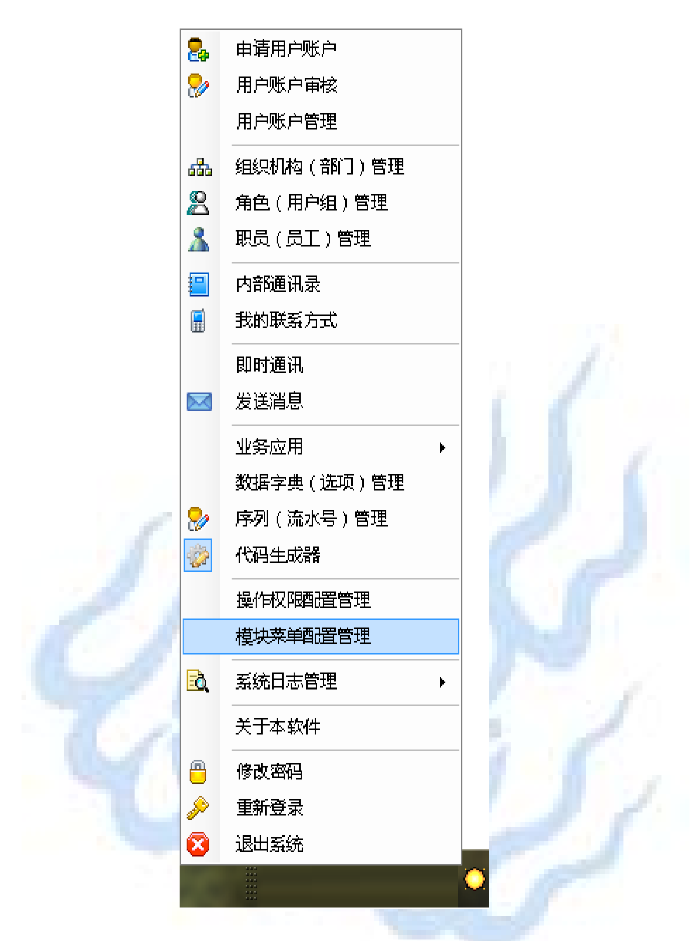


图 11-1 模块（菜单）管理菜单

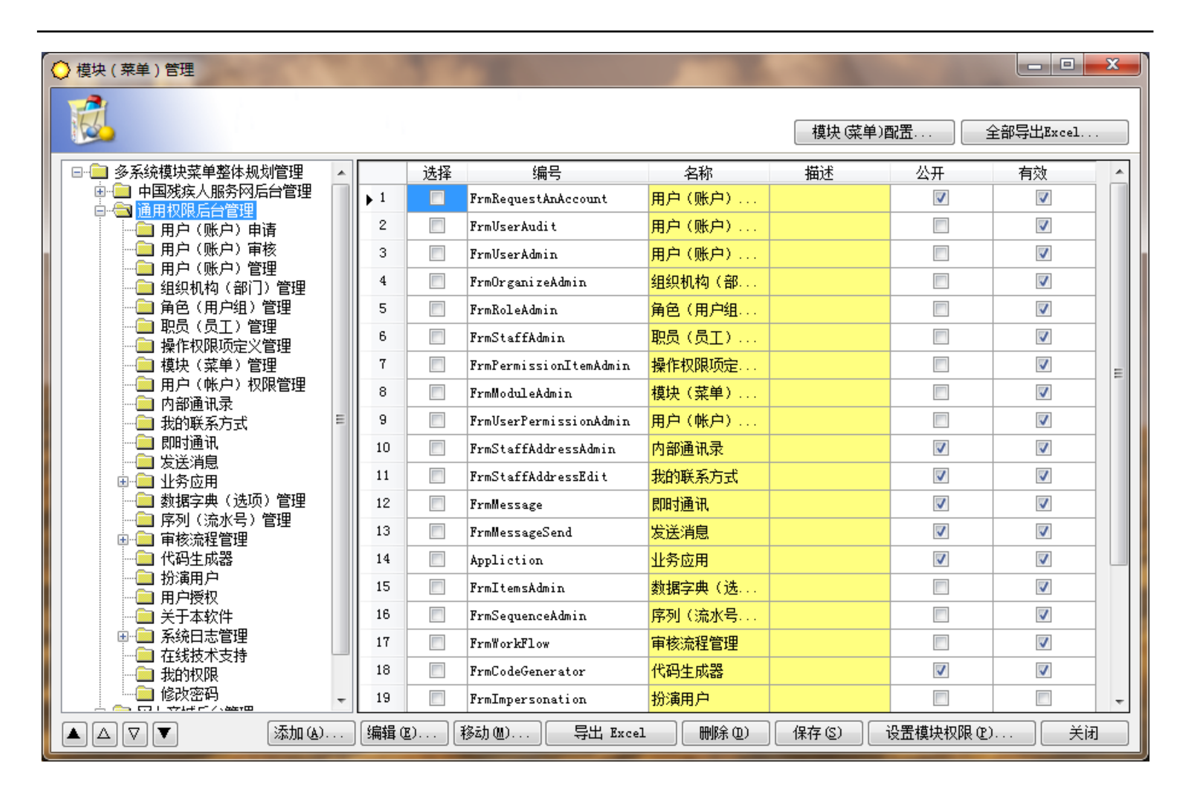


图 11-2 模块（菜单）管理窗体

在开发某一个软件项目、或者软件产品的过程中，往往菜单的名称、菜单的摆放位置、菜单的分组等等会经常发生变化或者进行调整，这时若菜单管理模块没有灵活的拖拽功能使用起来会很笨拙、设置是变成一种痛苦，所以此模块提供了灵活的拖拽功能，而且修改菜单名称、调整摆放顺序等操作时都提供了便捷的操作方法。

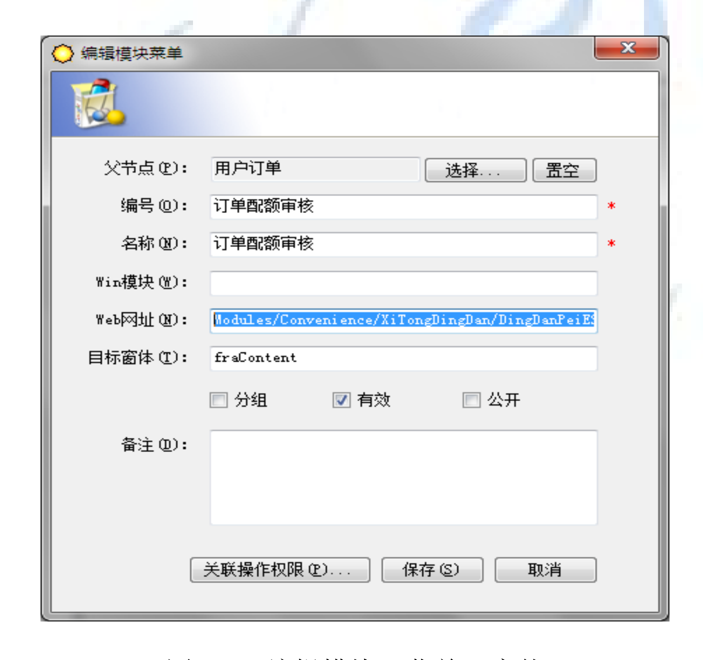


图 11-3 编辑模块（菜单）窗体



在此模块中，可以添加、删除、修改、拖动菜单位置，还可以导出菜单数据。可以将操作权限与模块（菜单）关联起来，有什么操作权限可以访问此功能模块的关联关系。

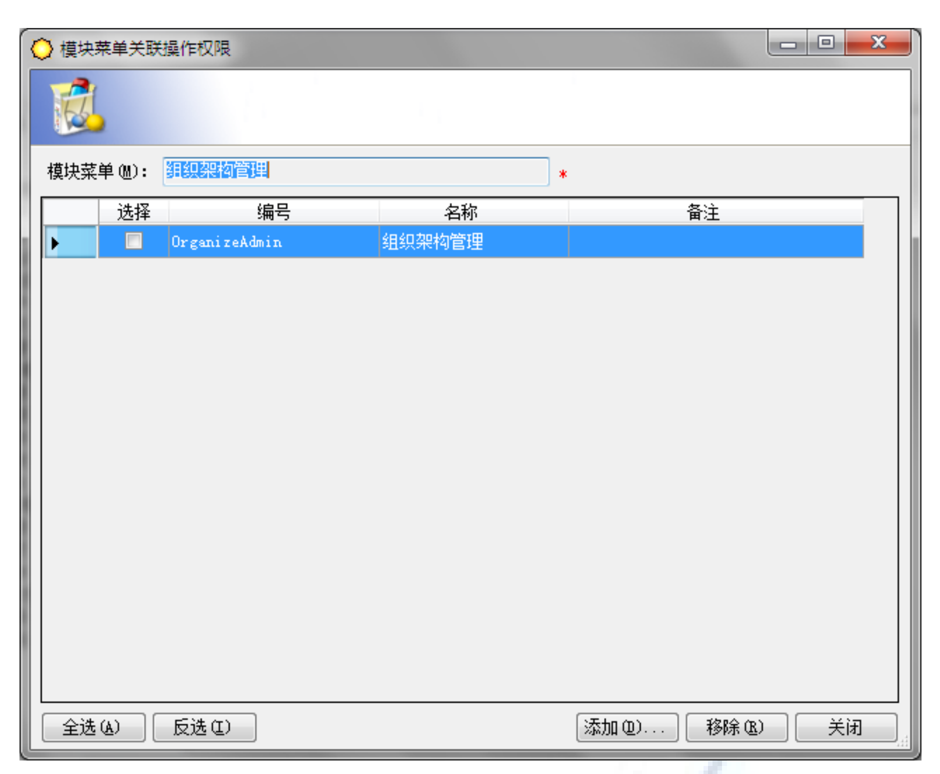


图 11-4 模块（菜单）关联操作权限窗体

上图中的含义为：若用户有“组织架构管理”权限，那就可以访问“组织架构管理”这个菜单。当然也有可能是由于用户有“P_A”权限，所以允许访问“M_A、M_B”模块（菜单），操作权限与允许访问的模块（菜单）之间是多对多的关系。

哪些模块是停用关闭状态，可以设置模块的“有效”状态，停用的模块会在整个系统中被屏蔽掉，不会显示在管理系统里。例如不需要职员（员工）管理模块、不需要日志功能、不需要组织机构（部门）管理功能模块等时都可以进行停用设置，将这些模块无效状态去掉就可以达到目的了。

第12章 操作权限项管理

操作权限项管理模块是本系统的核心模块，在此模块里进行操作权限的定义、配置、分类、管理等维护工作，可以在本模块进行操作权限添加、修改、删除、拖动、停用操作权限等功能，进入本模块的菜单如下图。

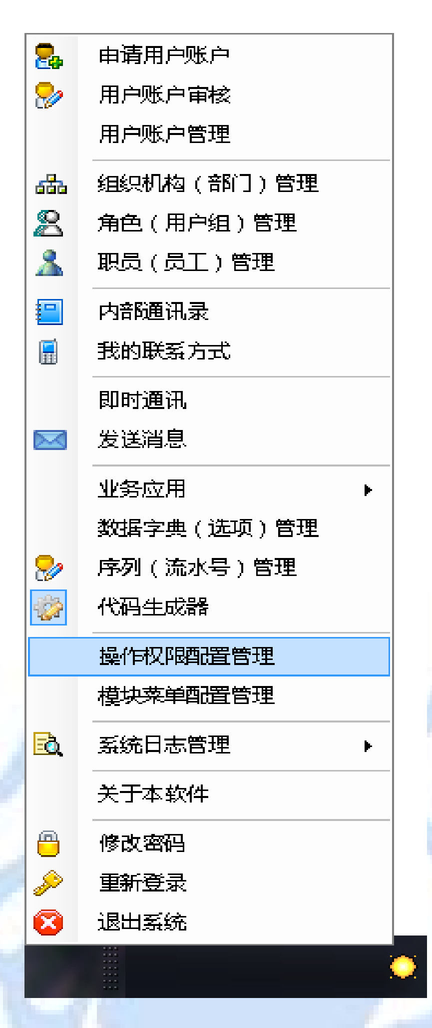


图 12-1 操作权限管理菜单

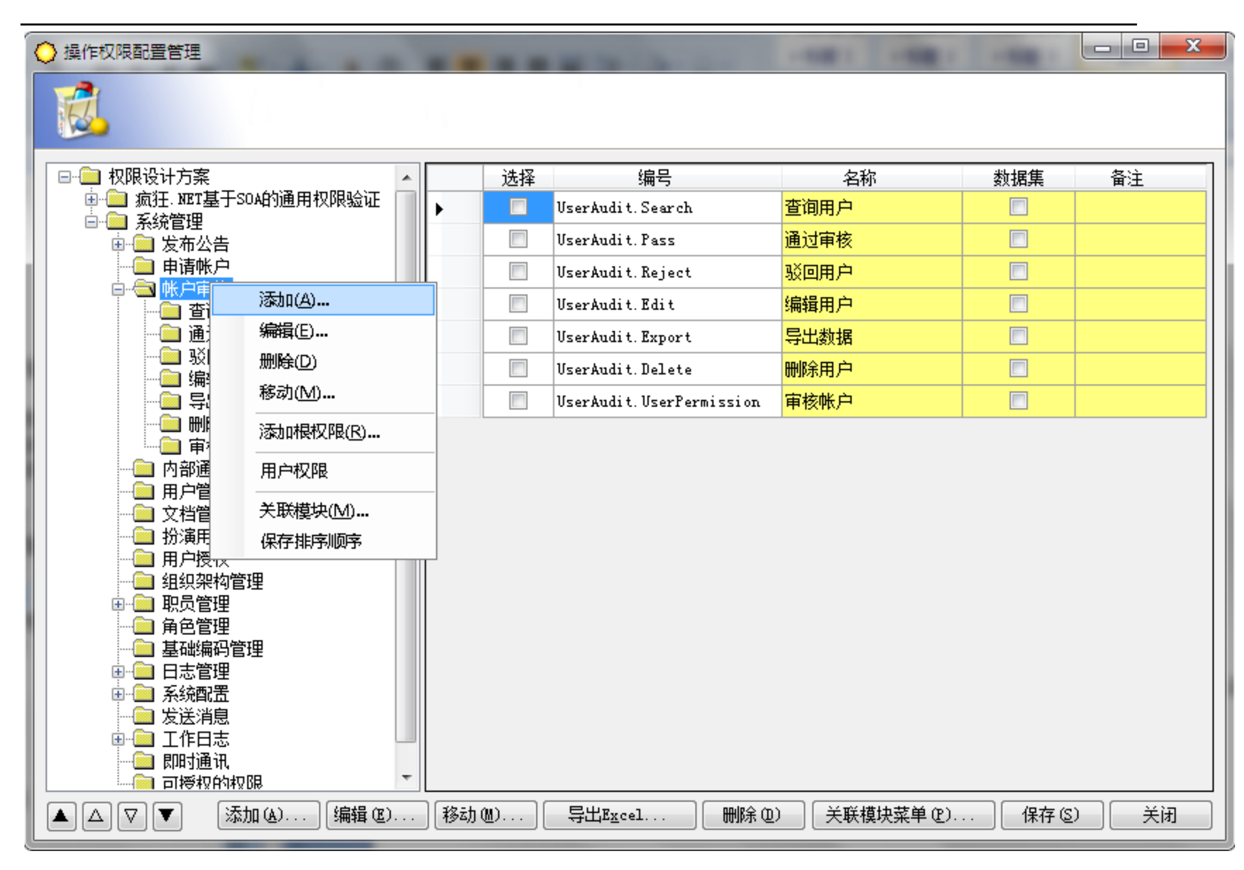


图 12-2 操作权限管理窗体

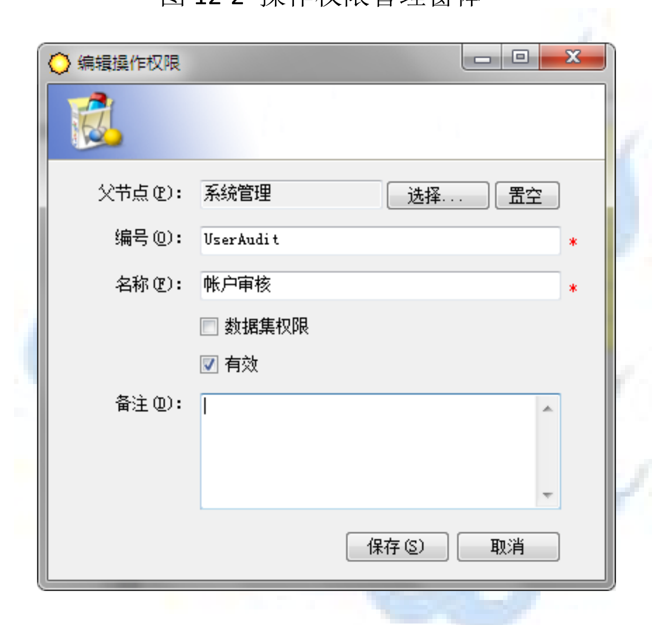


图 12-3 编辑操作权限窗体

可以在本窗体中集中管理操作权限，只有系统管理员能有相应的操作权限，操作权限的编号不能重复，在整个系统里需要有唯一性，名称可以重复，操作权限可以任意分类，支持自由拖动整理。

在前期设计和开发设计阶段，往往权限设计反复推敲、推倒、重新设计、昨天刚设置了某一权限、今天又觉得不合理，应该推倒，后天又觉得应该与哪个权限合并才对，没过几天又觉得这个权限又有必要分开来才对，经过反复的不断推倒从来，经过反复重新设计整理推敲过程，才能最终形成比较合理的权限管理方案，这个反复折腾的过程要求权限规划管理模块要有非常灵活的操作性，能经得起客户的反反复复，快速有效的适应客户的整体调整才可以，此权限管理也是经历了多个实际项目的多次实践和不断完善总结出来的结晶部分。

数据列的访问权限，其实也是属于一种特殊的操作权限，也是属于操作权限的一种范畴，例如一个权限就叫“访问 A 表的 C1 列权限、访问 A 表的 C2 列权限”就可以了，没必要设计得过于复杂，例如建立一个“拒绝访问 A 表”的权限，谁拥有这个权限，就不允许访问 A 表就可以了。

数据集权限是表示拥有某个权限，但是这个权限需要区分数据的作用范围，虽然这个帐号有管理权限但是只能管理 A、B 部门的数据，不能管理 C、D 部门的数据等，本系统的数据集权限，主要是用组织机构数据进行过滤，提供多种可二次开发的接口功能，达到按客户的实际情况进行个性化的数据过滤的要求，并不是所有的权限都能设置数据集权限，只有这个权限被设定为是数据集权限，才可以设置相应的过滤条件，数据集权限设置参考下图。



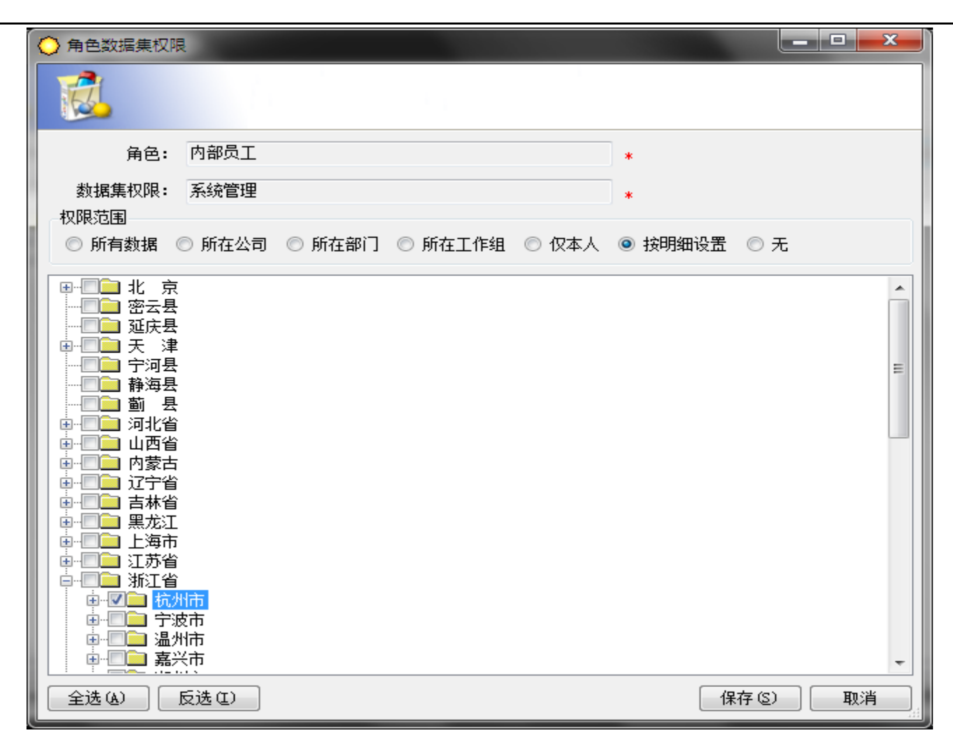


图 12-4 操作权限的数据集权限设置效果图

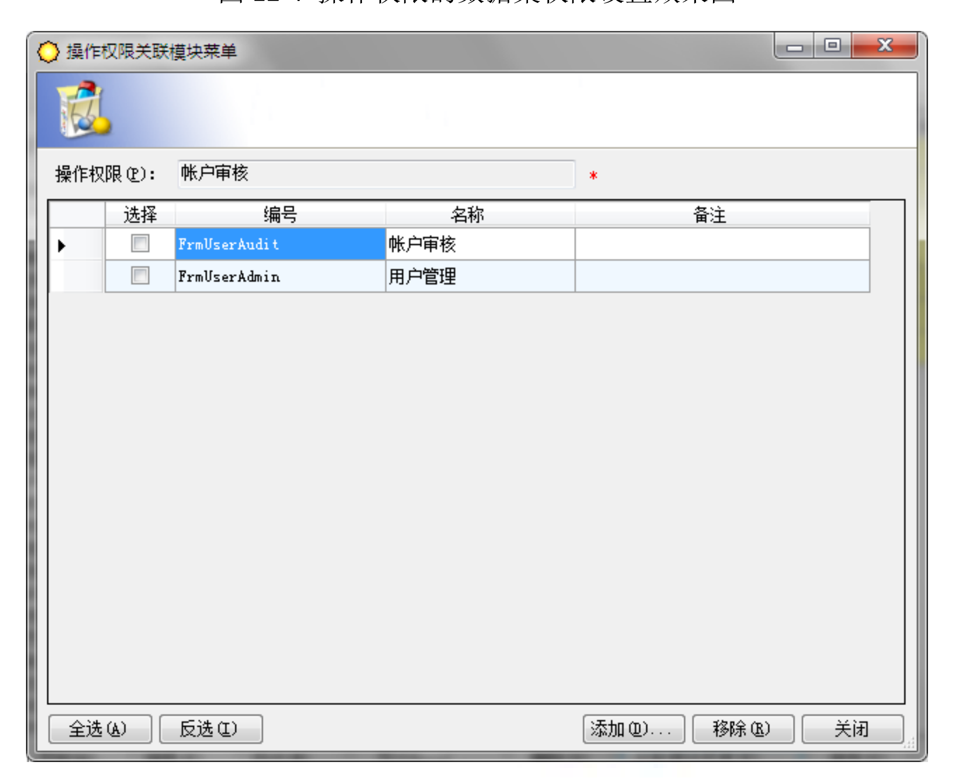


图 12-5 操作权限关联模块（菜单）窗体

操作权限关联模块（菜单）的主要用意是：当前的用户（帐号）已经拥有了某个权限，那有这个权限的用户（帐号）又允许访问那些菜单模块呢？是否可以自动进行关



联关系设置，不需要重复进行模块访问权限？主要是为了解决以上的智能关联问题，将权限与模块进行了关联设置。

操作权限于模块（菜单）之间设置了关联关系，就表示有了某个操作权限后，自然可以访问某些菜单模块了，不需要进行单独的模块访菜单的访问权限了，这样可以题高整个系统的智能逻辑关联关系，使得分配权限更加合理、简洁高效。





第13章 用户权限管理

每天都在发生变化、随着公司内部人员的变动，系统中的用户的角色、权限经常也会发生变化，这时候需要有一个灵活的权限设置功能，可以快速便捷的进行权限的重新划分，权限的重新设置分配等等。

往往开发系统，在上线前就应该做好帐号数据的整理、权限的规划、角色的定位，每个帐号到底应该有什么权限、到底能访问那些系统资源、到底能做哪些操作，都应该进行详细的规划，在系统实施、测试过程中就应该完成这些正式上线前的准备工作，若一个单位有几百个人，一个系统有几十个模块，这个工作也应该明确指派给一个人来负责比较好，由这个人统一负责规划、设置权限，这样不会乱成一团。

有个稳定的权限配置工具，在项目开始阶段，就可以安排一个人来负责把客户的组织机构数据、用户数据、权限划分，菜单的规划等等，都同步做好，这样可以明显提高并发工作能力，可以精力集中解决业务系统问题上，有效利用有限的人力资源，同时系统的功能模块，又需要依赖一些基础性的测试数据，这样整个开发、实施工作可以同时进行，又在开发前已经把基础数据都准备好，对接下来的开发工作还是很有帮助的。

同时有个稳定的管理工具，不会在这么基础性的东西上反复浪费时间，能提高工作效率，而且客户从项目的一开始就看到了有一些实质性的进展，也会放心一些，也可以明显减少在项目刚开始开发，就进入了高强度的劳动状态，需要处理的事情很繁琐，开发人员更容易提前进入疲劳期，项目失败的概率也高了很多。

第14章 序号（流水号）管理

序号（流水号）管理菜单如下图，主要起多种数据库的兼容性主键生成、数据表的排序字段的生成、单据流水号生成功能等。

本系统的多数据库支持功能，主要是依靠了底层的序号生成器的兼容性主键思想来实现的，例如不同数据库的有不同的主键生成思想，例如 Oracle 数据库用序号功能产生主键的较多、SQLServer 数据库用自增量做为主键的产生方法较多，但是这些方法都有一定的局限性、若遇到数据库移植问题，需要修改很多代码才能达到目的。

用了序号（流水号）管理组件，就可以很轻松的做出多种数据库兼容的主键产生方法、这样有数据库移植问题时，代码的修改量会很少，甚至有可能是是没有代码修改量。

其中单据流水号生成功能，支持灵活的二次开发接口，可以按某天、某月、某年，自动生成某个单据号的生成方法，也能支持并发请求下产生唯一单据号的功能，例如年月日_001、年月_001、年_001 等等形式的单据号形成。

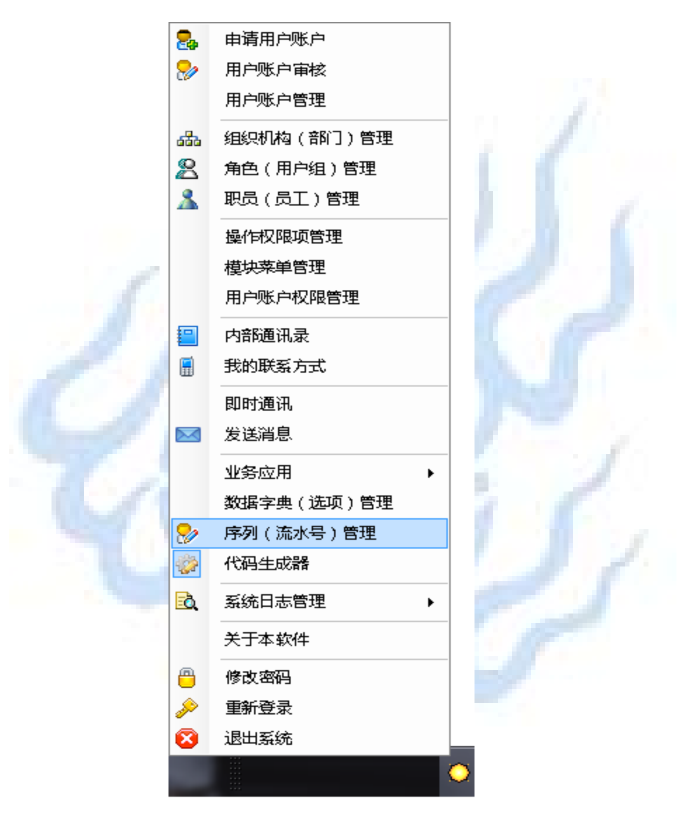


图 14-1 序号（流水号）管理菜单

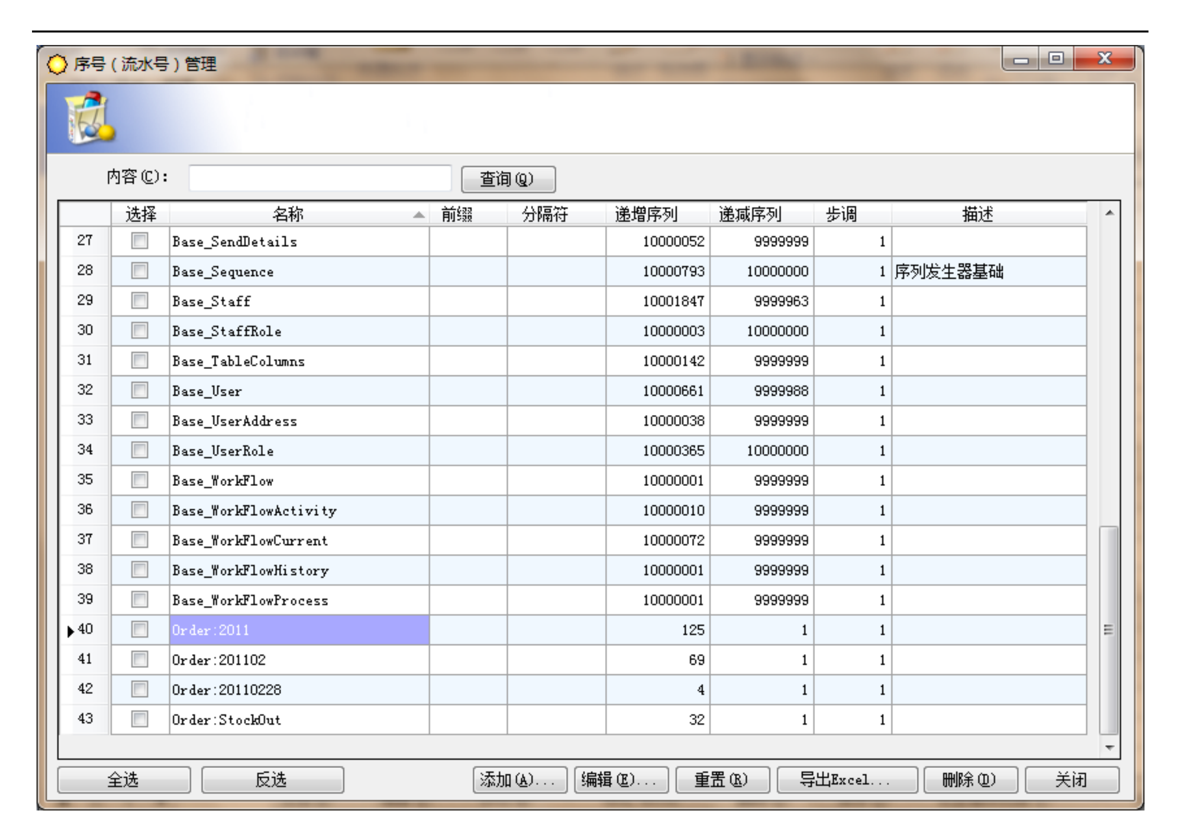


图 14-2 序号（流水号）管理窗体

可以在序号（流水号）管理窗体中，人工干预当前的某个单据序号，可以删除某个序号、修改某个序号，并支持导出数据处理。

重置功能是为了在将开发测试阶段产生的订单号清除掉，方便重新开始生成订单号为目的设计的。

增序号、减序号，还是为产生排序码设计的，分别对应最大的排序码、最小的排序码，为了快速实现把数据排序到最前面，或者把数据排序到最后面时用了这个 2 个数据值。

其实本功能模块的用法，很接近 Oralce 数据库中的序号功能，只是提供了更加便捷的用 C#可以直接调用的功能接口。



第15章 系统异常情况记录

系统日常维护管理和二次开发过程中发生的所有异常信息都会被进行记录，方便开发人员排查错误，可以成为修正错误时成为有力依据。

系统中发生的异常记录可以逐条打印清单然后发送邮件给相关的开发人员，等到系统配置上线后，还可以进行统一的整理和清理工作，方便监控在日常使用过程中发生的异常情况。

客户端发生的异常被记录在应用程序相应的文本文件中；服务器端发生的异常情况会被记录在相应的数据库系统里。

进入系统异常情况记录的进入的菜单如下图所示。

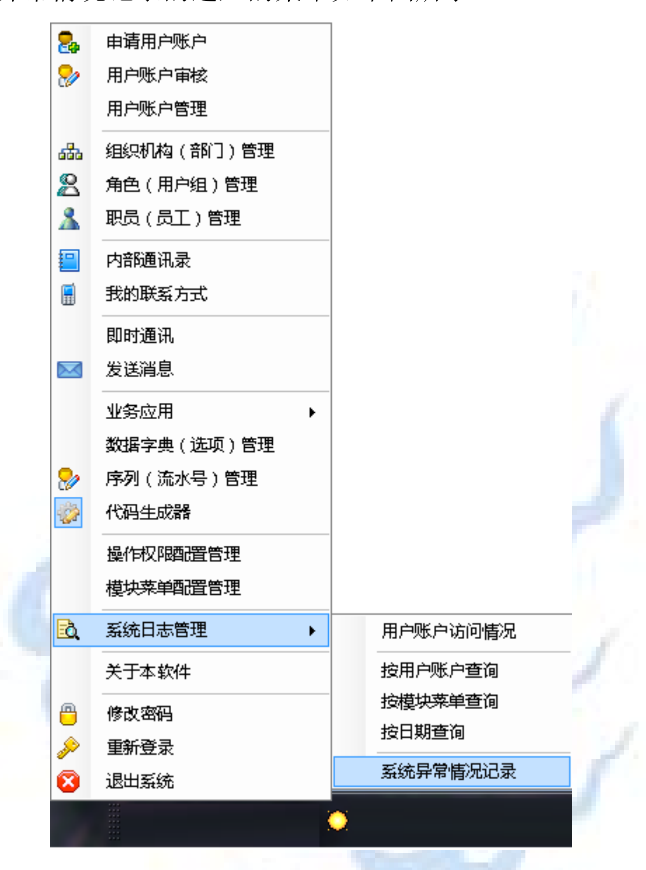


图 15-1 系统异常情况记录菜单





图 15-2 系统异常情况记录窗体

系统中的异常信息可以导出到 Excel，方便反馈给开发商，也可以阶段性的清除异常信息，这样最新发生的异常信息可以很方便的觉察得到。



图 15-3 系统异常情况记录详细信息

可以将异常信息逐个打印出来，传真给开发商或者直接通过邮件方式发送给内置的错误反馈邮件中。

第16章 关于本软件

本功能主要是为了显示软件的版权信息内容，例如软件名称、版本号等信息。这些版权信息可以在相应的配置文件、源代码中进行配置修改；先有功能可作为开发者的参考，进行简单的修改达到重复利用的目的。

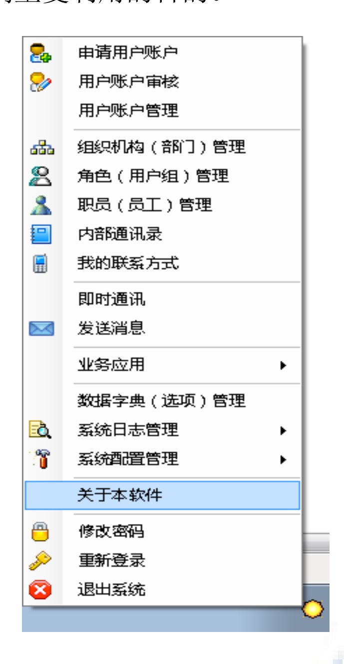


图 16-1 关于本软件菜单



图 16-2 软件版权信息

版权页面上点任意鼠标键就可以退出页面返回信息管理系统主页，此页面也可以起到客户了解管理信息系统的当前版本号，向开发公司、开发者反馈意见、建议时的重要的参考信息之一。

第17章 修改密码

修改密码模块主要是对用户（帐号）密码进行修改操作，在修改密码界面必须输入原密码、新设置的密码后按确定即可完成对密码的修改，若原密码输入错误将无法进行新密码的设置工作，这也是为了防止盗取不在操作岗位上的员帐号的密码的一种保护措施。

新密码需要确认主要是为了防止用户输入错误导致不能正确确认自己的密码，重复输入两次新密码就是为了让用户确认，绝对是没错的。

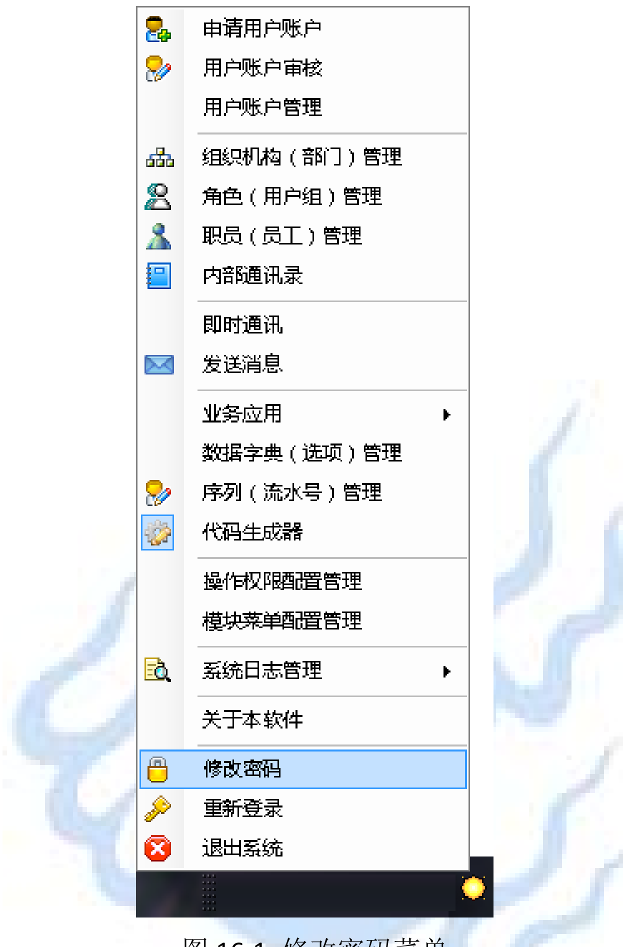


图 16-1 修改密码菜单



图 16-2 修改密码窗体

若系统允许设置空密码、那新密码、确认密码可以不输入，只输入原密码将新密码设置为空，直接按确认就可以了。

系统配置为不允许空密码、那原密码、新密码、确认密码都是必输入项目，会有红色 *标识。

系统配置为不允许空密码、登录系统时密码也是必须输入的，会有红色 * 标识，如下图。

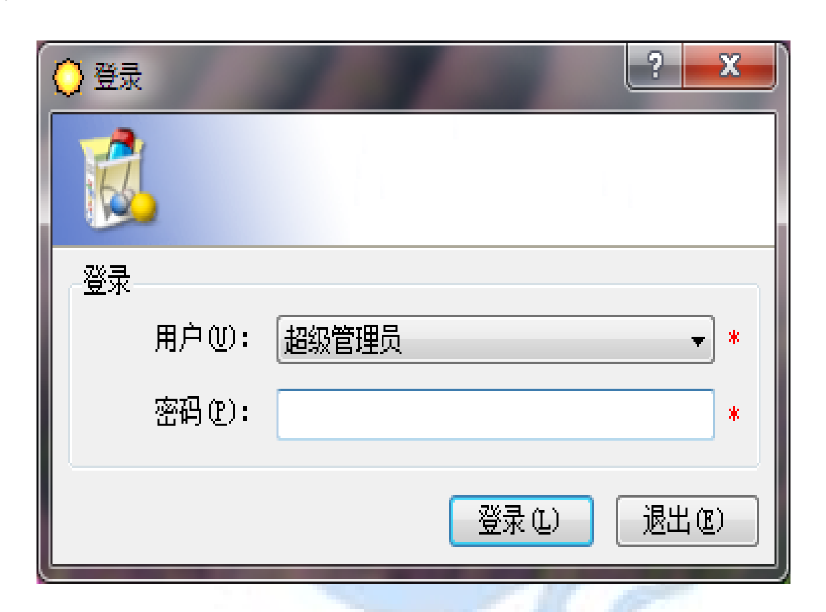


图 16-4 登录窗体



第18章 重新登录

特别是在软件系统开发、测试过程中经常需要在几个帐号之间快速切换，若每次都退出系统重新登录效率低、操作的步骤也多、不连贯。



图 18-1 重新登录菜单

重新登录系统时，技术层面需要处理的事情主要有：用户的在线状态变化和用户的权限发生变化以及用户的菜单显示方式会变化，用户的即时信息收发情况会发生变化，取消登录不应退出当前的信息管理系统应该返回原来操作员的操作界面。

对比一下登录系统界面与重新登录系统的界面，一个是“退出”一个是“取消”按钮，他们还是有比较大的区别的。

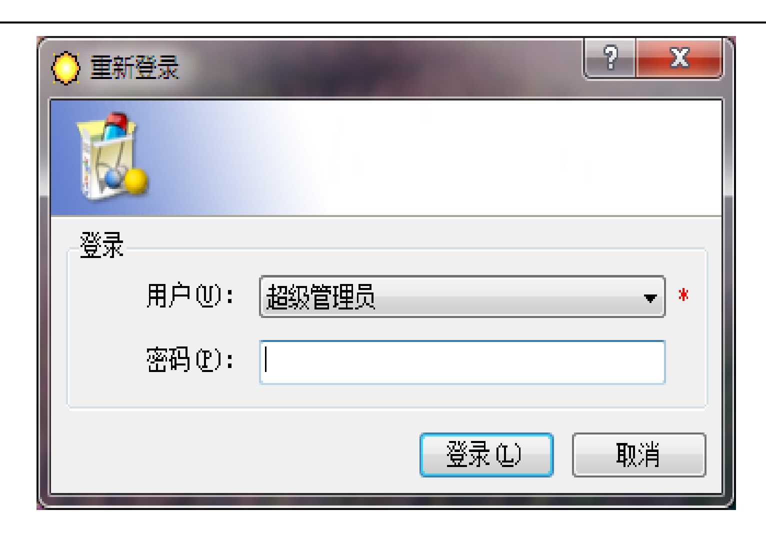


图 18-2 重新登录页面

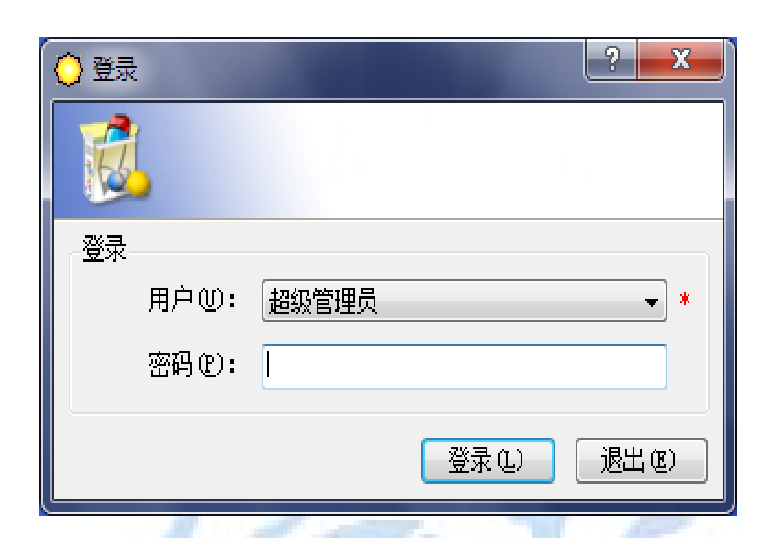


图 18-3 登录窗体

点击菜单中的重新登录菜单项目，返回系统的登录界面，通过登录页面可以重新选择用户名，输入密码进行登录动作而不用退出当前的管理系统。

若用户的权限设置发生过变化或操作员用户（帐号）信息等发生变化后建议重新登录系统，重新登录时会默认显示当前的用户名，直接输入密码就可以重新登录操作。

第19章 退出系统

退出系统后，应该更新用户的在线状态，已退出系统的用户设置为离线状态。每次退出系统时都进行提示确认，若客户觉得操作麻烦、还可以进行个性化的配置，直接退出系统不进行确认操作。



图 19-1 退出系统菜单

点退出系统菜单，会弹出确认对话框。

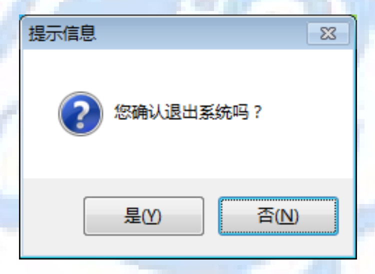


图 19-2 退出系统确认页面

默认情况下光标停留在选项“否(N)”上，点“是(Y)”选项就可以退出当前登录的管理系统，按“否(N)”选项退回当前的管理系统。



第20章 参数服务

第21章 权限基本设计理念

大概在接近 10 年前做 VB 版本的 ERP 系统，其中的权限分配也完全是类似下面的设计思路，分若干个模块、有新增、删除、编辑、删除、扩展 1、扩展 2、扩展 3 等操作权限定义，当时也很佩服设计类似权限系统的人，感觉设计这个的人真的是很有水平，甚至在后来的 N 多业务系统里也进行模仿，也做过类似的权限判断优化的存储过程程序等，当年也非常有成就感，呵呵。

No.	菜单名称	全选 ☐	查看 ☐	新增 ☐	编辑 ☐	删除 ☐	打印 ☐	审核 ☐	作废 ☐	结束 ☐	扩展 ☐
1	⊟工程项目管理系统	☐	☑	☑	☑	☑	☑	☑	☑	☐	☑
2	⊞项目信息	☐	☑	☐	☑	☐	☑	☑	☑	☐	☑
8	⊞进度计划	☐	☑	☑	☑	☑	☑	☑	☑	☐	☑
18	⊞成本预算	☐	☑	☑	☑	☑	☑	☑	☑	☐	☑
25	⊞双包管理	☐	☑	☑	☑	☑	☑	☑	☑	☐	☑
42	⊞单包管理	☐	☑	☑	☑	☑	☑	☑	☑	☐	☑
59	⊞材料管理	☐	☑	☑	☑	☑	☑	☑	☑	☐	☑
98	⊞质量管理	☐	☑	☑	☑	☑	☑	☑	☑	☐	☑
113	⊞安全管理	☐	☑	☑	☑	☑	☑	☑	☑	☐	☑
120	⊞工程资料	☐	☑	☑	☑	☑	☑	☑	☑	☐	☑
126	⊞工程资金	☐	☑	☑	☑	☑	☑	☑	☑	☐	☑
219	⊞内部结算	☐	☑	☐	☐	☐	☑	☑	☑	☐	☑

当还没深入开发时也感觉不出来其中的奥秘，也体会不出来否好用还是不好用，就感觉看着是很强大，足够强大，还可以无限扩展一样。

随着自己着手开发的信息化业务系统越来越多、经验积累也越来越多、各种体会也越来越深刻，从最近 5 年左右开始，逐渐发现这个权限设计存在蛮多问题，有很多细节方面无法满足开发要求灵活性，先简单的总结一下其中的不足，虽然对付用用还是没问题：

1：例如有些模块，是没删除功能的也没结束功能的，例如打个比方“成本预算”不是啥模块都清一色“添加、删除、修改”，权限哪里有那么简单了。

2：无法进行灵活扩展，例如某一个人有人事招聘权限、面试权限、笔试权限、确定岗位权限、确定试用期期限的权限，那怎么扩展了？那不是没完没了了？

3：判断权限的函数相对繁琐，需要判断哪个模块上的什么操作权限？2 个参数才可以，相对于只进行有什么操作权限判断还是过于复杂。

4：无法简化、细化操作权限项，例如我想简化为 5 个权限，不想设置这么多权限时，就会麻烦一些。

由于存在以上几点缺点无法得到比较好的解决，后来权限管理的设计就进行了简化、使得更加灵活了，更加好扩展了。



操作权限的判断函数也变得非常简单，就一行代码就可以了，思路也简化了，灵活性也有了。

```
// 这些是操作权限，当前用户有什么相应的操作权限？
this.permissionAdd = this.IsAuthorized("RoleAdmin.Add");
```

一个好的权限管理系统，若刚开始就走错了方向，那接下来越走就越折腾了，很多时候往往是设计思想的一点点变化，可能会导致全局上的天大差别，有的软件越走越臃肿，越累。

其实很多时候，往往是差距那么一点点的小小的设计理念，小小的功能差别，就能绝对胜败。例如海湾战争期间，伊拉克的坦克也是坦克，老美的坦克也是坦克，他们很多功能都有，但是人家老美的坦克就能多打几百米，那就整个战场上可能就由于有几百米的差距，带来惨痛的损失，在软件开发上也是一样的，很多功能上的一些细小的差距很可能导致了，整个软件系统就是没人家的好用。

第22章 多系统权限集中配置管理

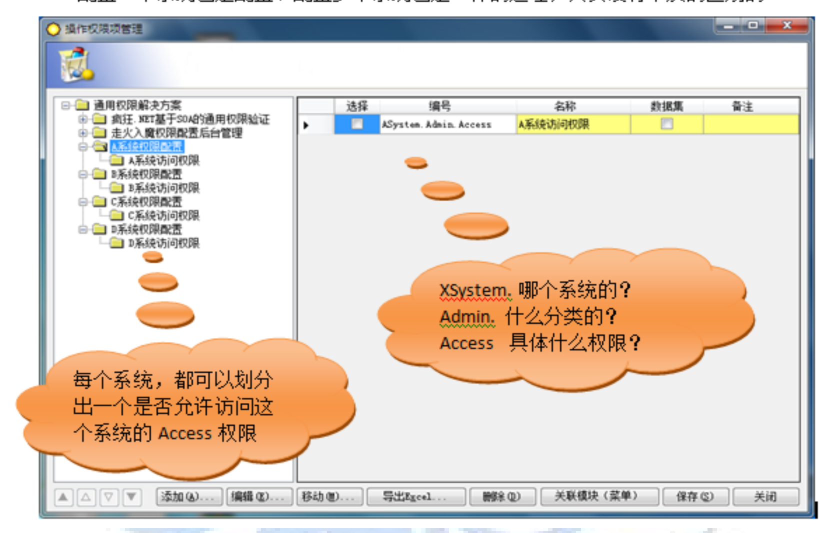


图 21-01 配置管理器

权限分配的参考页面为：

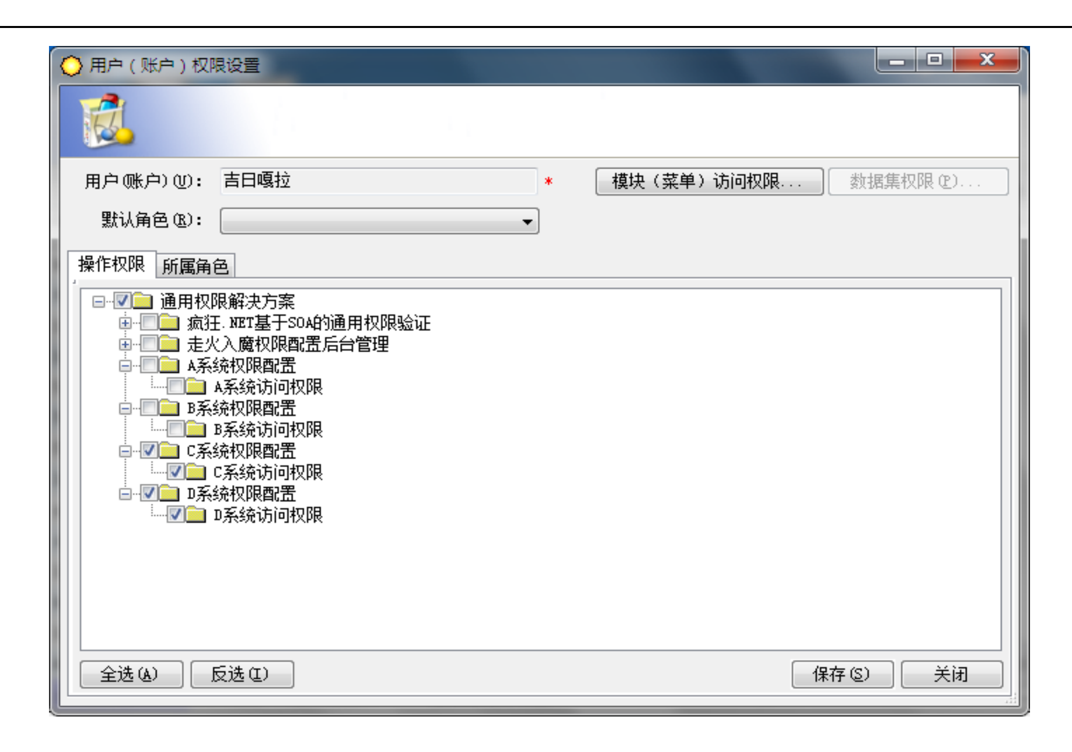


图 21-02 权限分配参考页面

用户角色的关联，给角色配置权限的参考页面为：

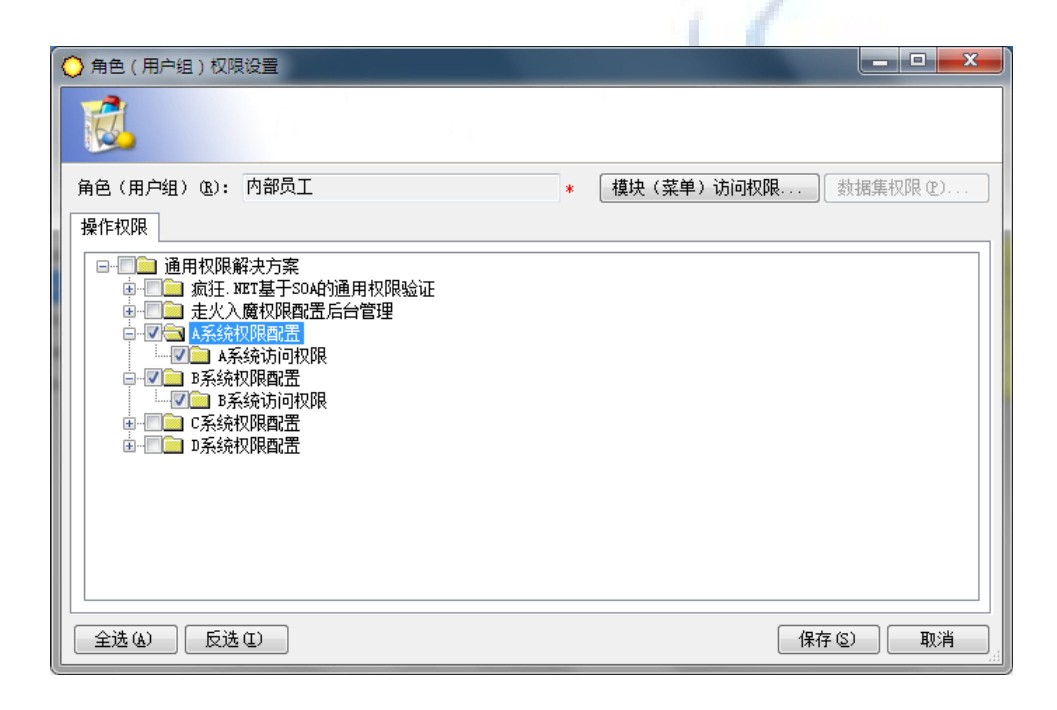


图 21-03 用户角色关联配置



有灵活的配置功能，就哪些角色可以访问哪些系统，哪些用户又归属那些角色，哪些用户又可以访问哪些系统等等，都可以通过配置就可以达到目的了，不用修改程序源码，随时随着客户的需求变化，随时可以进行配置，很灵活，能经得起客户的折腾以及用户需求的变化。

系统设计充分严禁可以按自己的需求灵活配置管理，想怎么管理就怎么管理，想怎么设计就怎么设计的管理工具，还是没那么容易的，经得起考验的工具，才有重复利用的价值。



第23章 自由行政审批组件

几乎没一个公司什么事情都一个人说了算，而是需要大家一起协商、协调，需要走一个流程的，这也是管理软件诞生流程审批组件的原因。

程序员请假

递交请假、
进入审核流程

审批流程组件的优点是请假流程等单据的审核过程不是固定的，是可以按客户的实际工作需要灵活配置的，是非固化的灵活的商业逻辑

开发主管审核

人事主管审核

总经理审核

审核流程结束
请假单生效

图 22-01 自由行政审批流程

大概在2002年左右在公司的门户系统 EIP，其中需要一个行政审批流程组件，当时是用 PHP 实现，数据库用了 MySQL。程序功能是按领导的要求，领导说要实现什么功能我们就把什么功能实现好，当时也没什么深刻的感受，反正也没耗费多久就做出来了，感觉也没啥的，用用还比较好用就可以了。



到了2008年，公司有一个几十万的 ERP 定制项目需要做，其中需要一个自由审批流程组件，那时候才意识到值钱的软件都需要审批流程组件，而且这个组件当时是由我来负责编写的，大概前后也写了1个月左右，又测试了1个月左右，才真正能用起来，而且当时还没能实现并行审批流程功能，一方面是经验不足，另一方面也是能力有限。

2010年又有一个 B/S 项目需要有审核流程，又耗费了2周时间把审批流程的底层数据库重新设计了一下，又耗费1周时间整体进行测试，终于把这个自由审批流程又重新整理一次，上了一个台阶，把上次没能实现的技术问题、审核时间限制、并行审核的技术问题解决好了，随着对日常工作管理上的要求的进步一加深，对工作环节上的审批的意义又有深入的意识了，管理软件的审批流程是太有必要了，审批流程成为管理软件的一个核心亮点。

2002年时，不知道这个审批流程管理组件有多么重要，也不知道其中的经济价值和客户的重视程度。2008年时，充分意识到了审批流程管理组件的重要性，而且要耗费很久才做出稳定的、拿的出手的软件，但是在编写程序时还遇到了一些技术难题与瓶颈。2010年时，把底层数据结构又重新设计一下、把底层代码又完善了一次，界面功能也进行了一次彻底的优化，组件使用起来更流畅舒服。

其实真正能重复利用的就是数据库设计 PDM 文件，这个也是业务领域的建模，用这个设计不管是开发 JAVA 的项目，还是开发.NET 的项目，都可以重复利用，有一定的参考价值。从设计上也可以看出一个人的思路是否严谨，是否设计考虑得充分，是否有一定的行业经验等。其实很多业务领域的建模都是有价值的，这才是软件开发中沉淀下来的劳动成果。



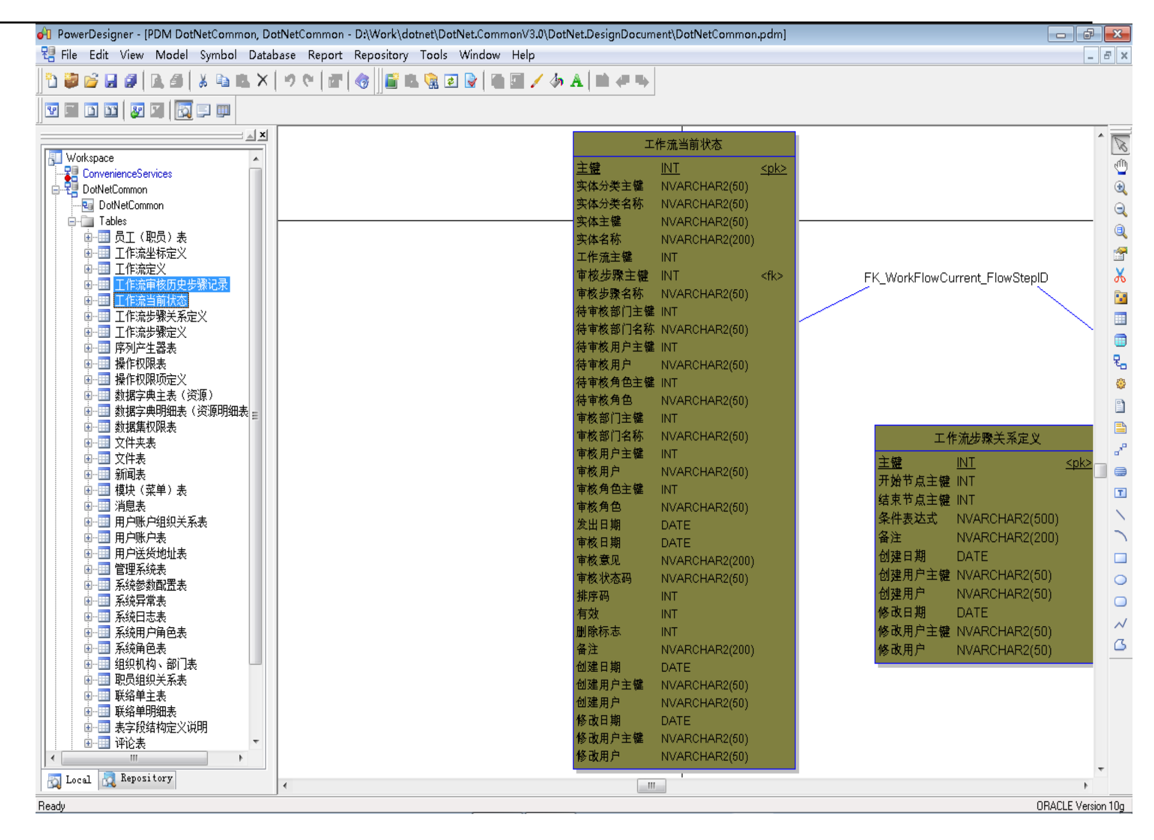


图 22-02 用户角色关联配置

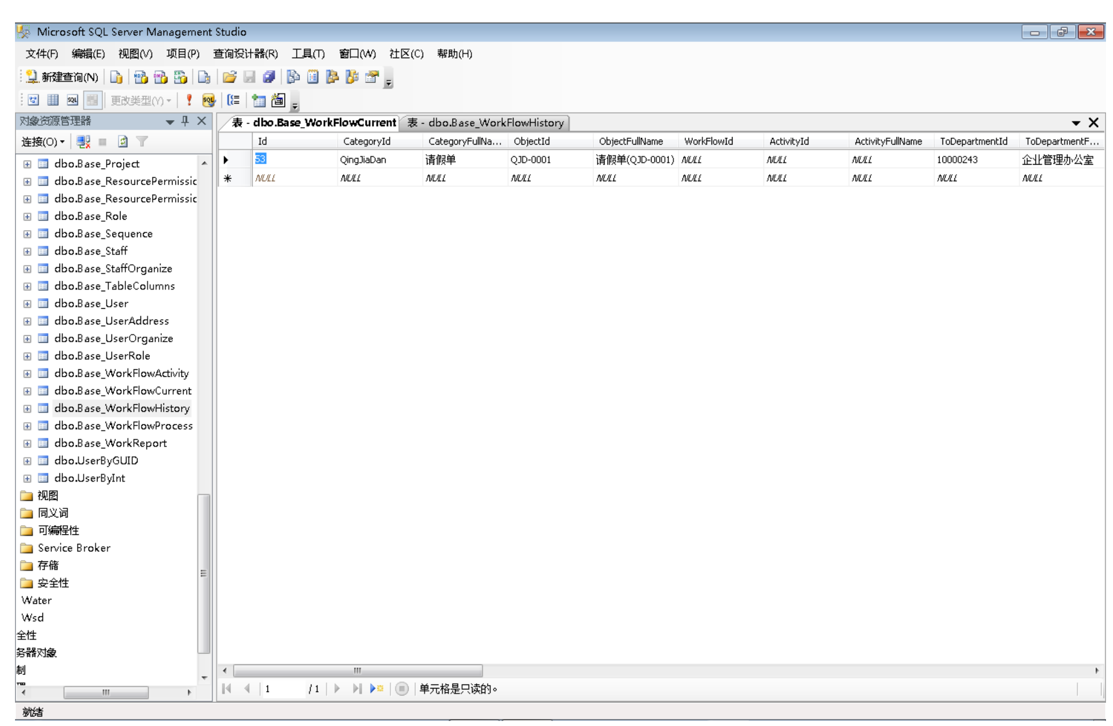


图 22-03 工作流当前审核状态表



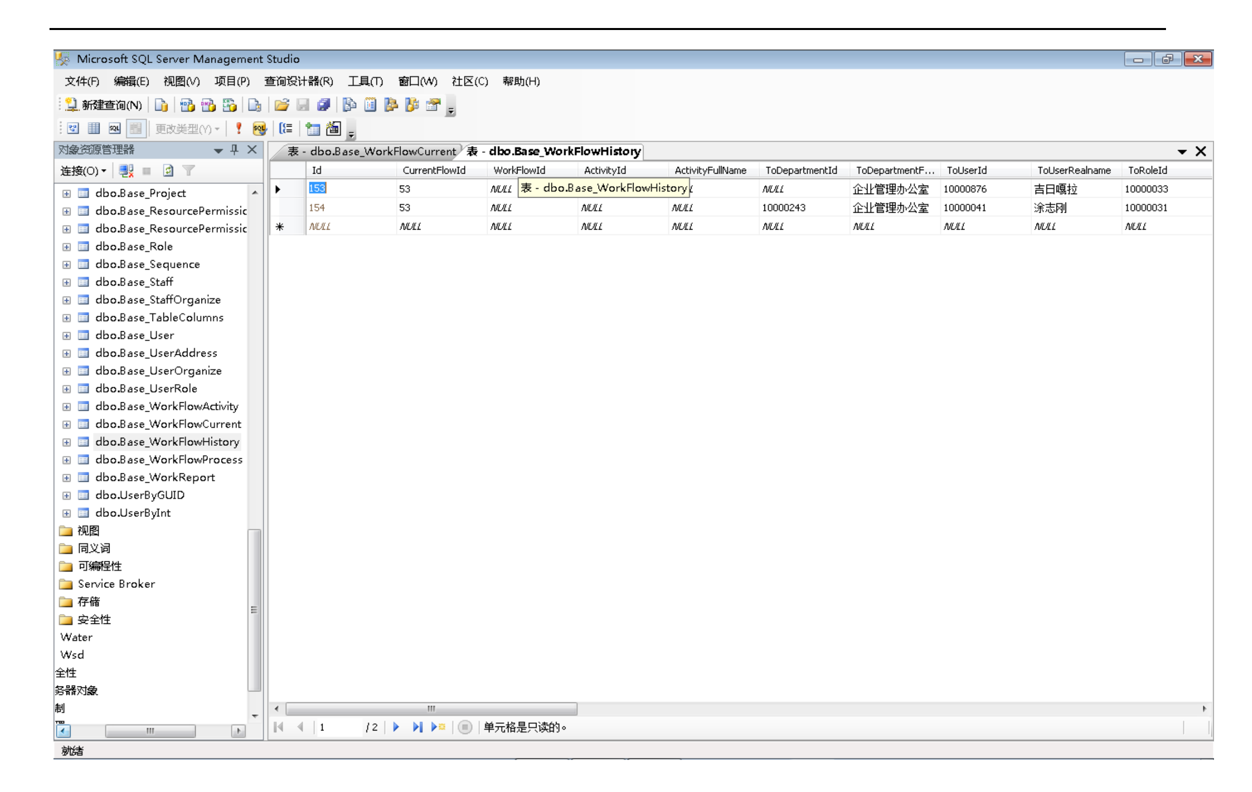


图 22-04 工作流审核历史步骤记录表

不管是 B/S 系统还是 C/S 系统，有些底层组件即所谓的商业逻辑组件是有很大的重复利用价值，这些代码里封装了很多业务逻辑，这些业务逻辑不管做什么系统都不可或缺的。这些严谨的有重复利用价值的商业逻辑代码积累越多，积累越严谨，越完善，将来可以赚大钱的机会就越多。因为等机会来了，软件可以很快被搞定。但编写这些严谨的商业逻辑代码往往会耗费几个月时间，一个月工资按1万计算，为了积累这些成熟的组件，至少都好烧掉几万元的成本，若采用了拿来主意，这几万元的开支就节省了。

从头实现一遍所耗费的时间往往比拿来看懂再改改的时间多很多，可能熟悉代码花费了1周修改代码也花费了1周，总的时间成本虽然有2周，但是自己从头全部写好可能需要2个月时间才能写好的。自己编写时总是会发现很多细节没处理好，写来写去，调试来调试去1个月就不知不觉中过去了。



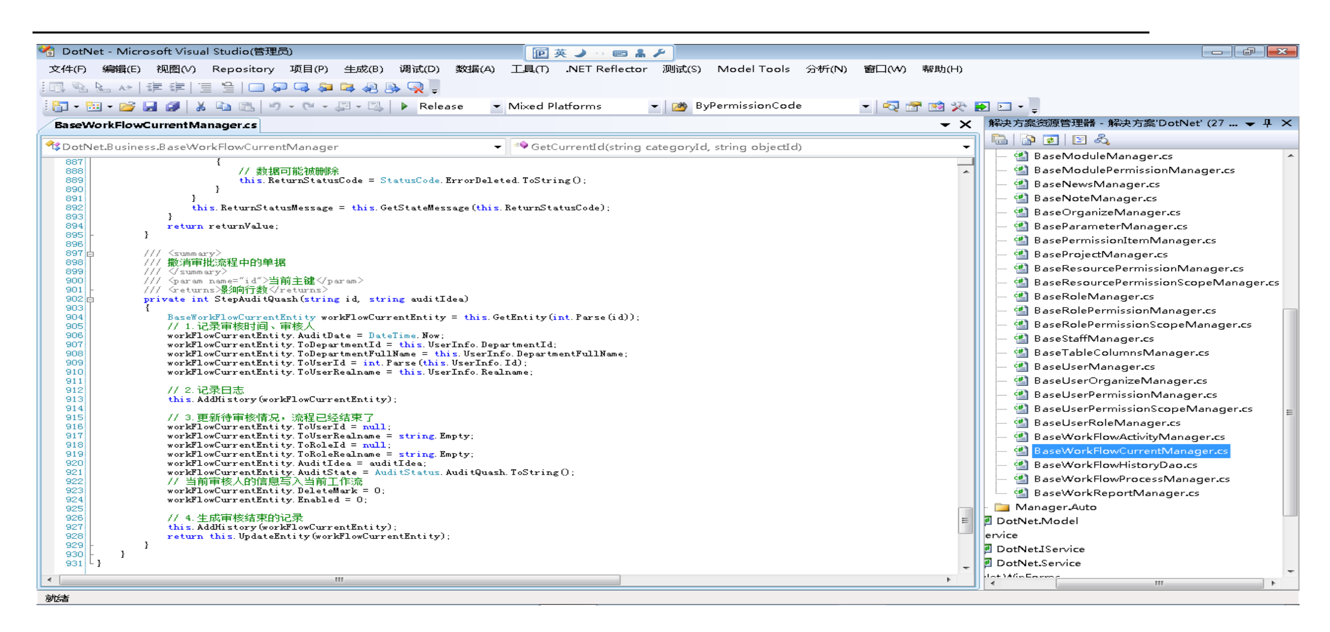


图 22-05 工作流审核历史步骤代码

接下来我们进入实战，以 C/S 的程序为例子，有需要时也可以做个 B/S 的审批流程例子，界面大致是一样的效果，因为都调用了相同的底层函数处理数据，数据设计也会是相同的，不会因为 B/S 而数据设计不同。我们假设有这样的一个自由工作流审批流程（假设请假流程）：开发部主管（吉日嘎拉） - 人事主管（涂志刚） - 总经理（胡况明）

用户帐号情况如下页面效果

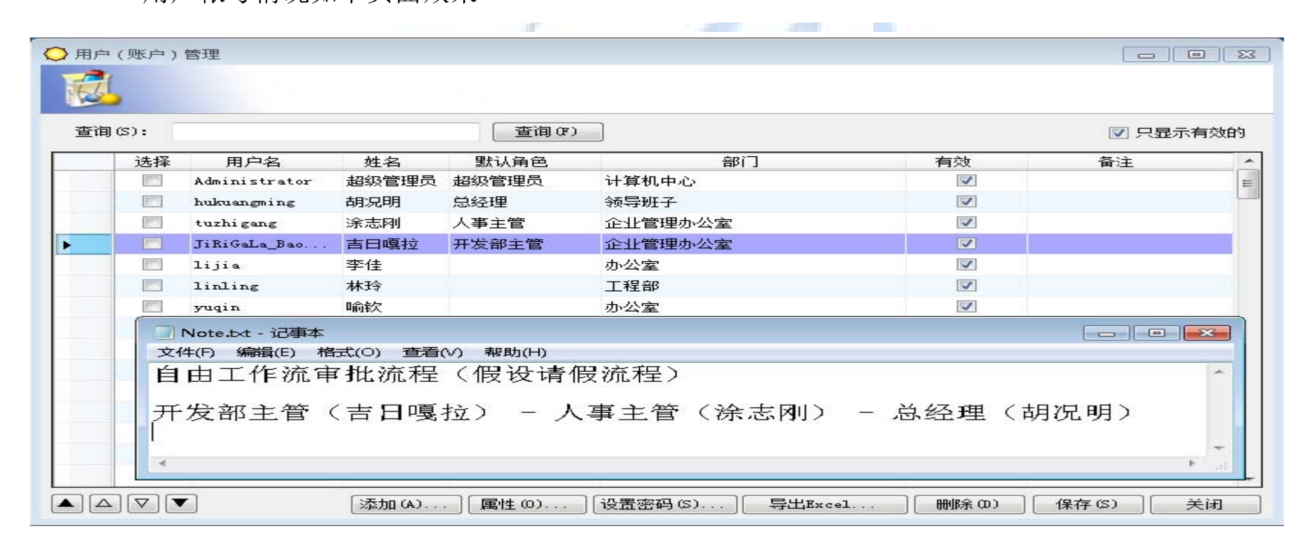


图 22-06 自由审批流程

首先我们需要启动这个审批流程，让一个单据进入审批流程环节。这里只是做了

一个示意图来显示请假单，也可以显示请假单的核心参数（当然可以是其他单据，不只限于请假单，还可以是入库单、出库单、付款单、借货单等等一大堆单据）。

Q&A

问：这里的表名、订单代码为什么要手动输入呢？有什么特别的好处吗？

答：这是一个单据的核心数据的意思，当然从数据库里读也可以，这个是唯一识别一个单据用的。例如请假单、入库单等,都可以走审批流程的,是我们的流程组件，需要知道有这4个信息,才能进入审核流程。

问：但是这几个让用户输入，有时候会让用户不知所措吧？就是不知道输入什么？

答：你误会了，不是让用户输入,这个页面应该是一个请假单页面（例如）程序只关心其中的4个参数,其他信息就没显示了,可能这4个参数还是隐藏的。

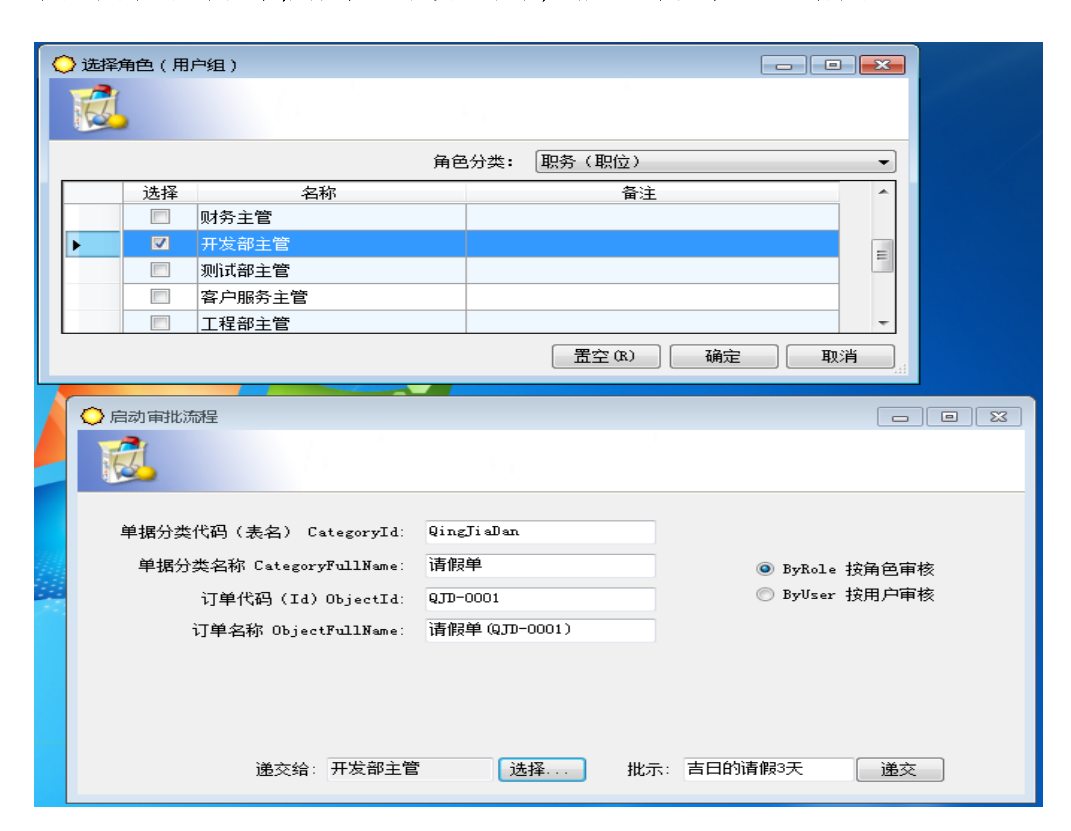


图22-07 自由审批流程配置

我们看看第一步审核流程的情况，先用吉日嘎拉帐号登录系统，看看会是什么效果

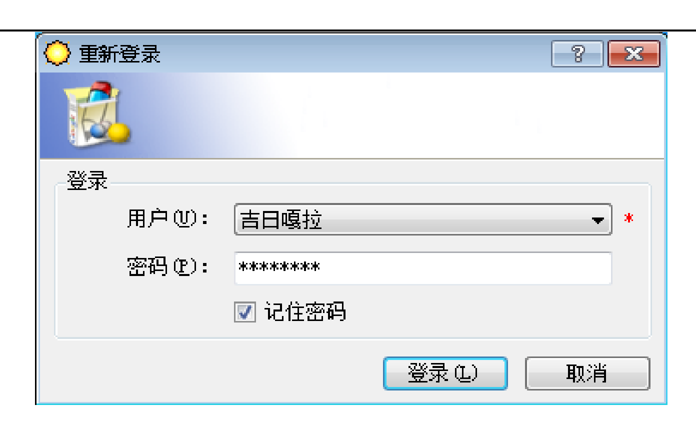


图22-08 审批流程登录

点待审批流程菜单，查看需要待审批的数据项



图22-09 检查待审批流程

的确显示出了待审批数据了，我们把请假单递交给下一个审批环节

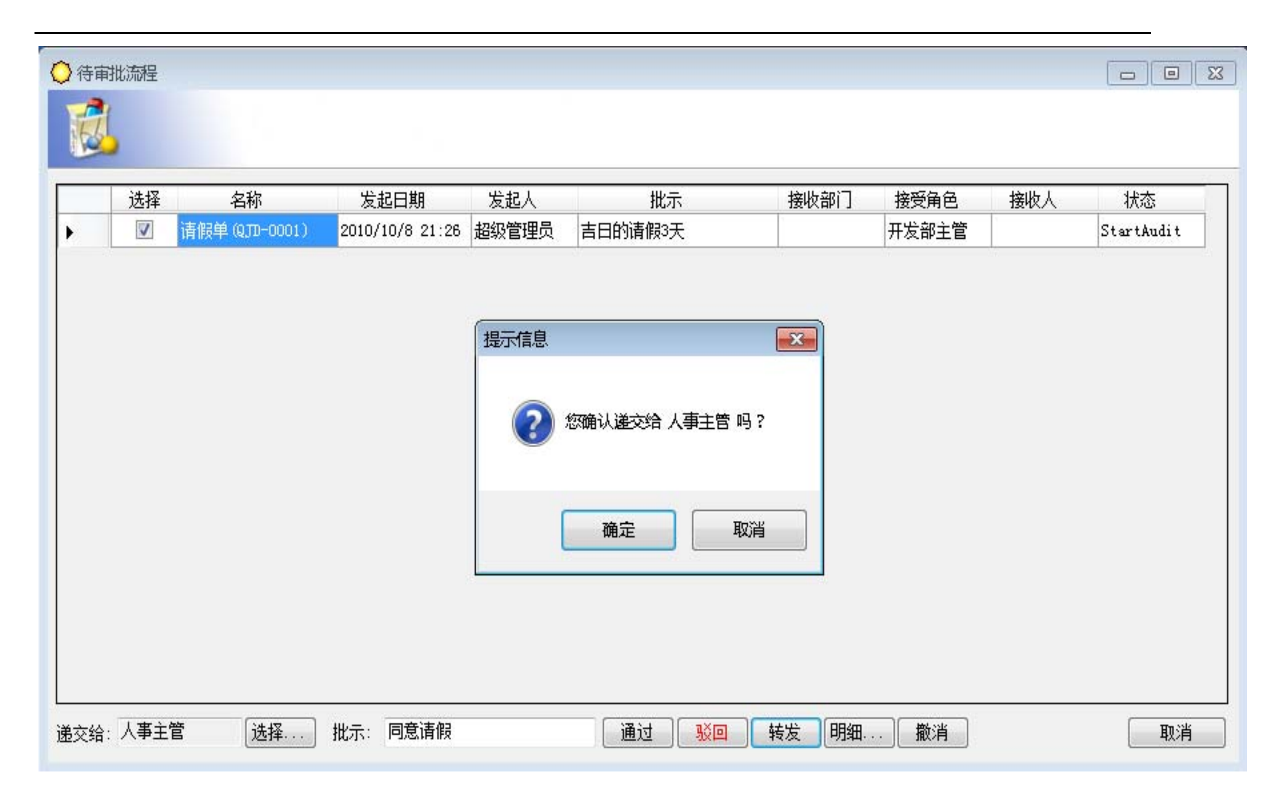


图22-10 进行审批

用下一个审批环节的用户进行登录

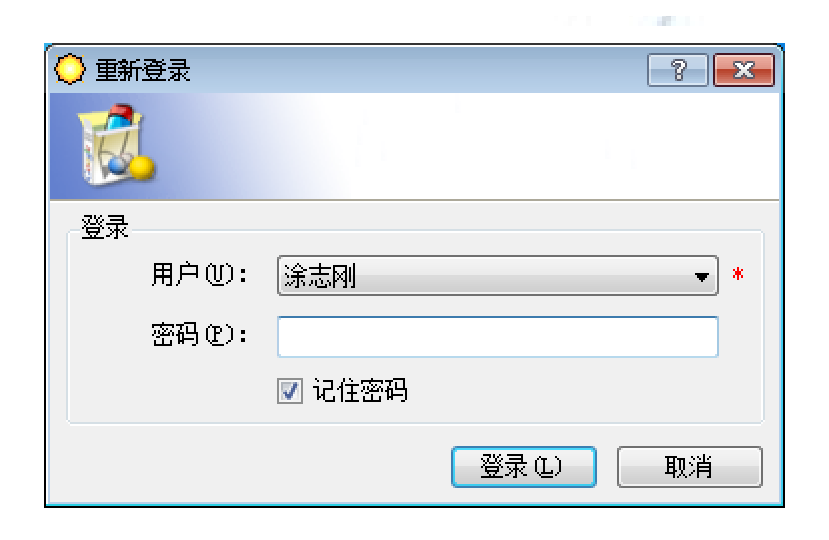


图22-11审批流程登录

正好是我们需要审批的数据显示出来了，同时我们审批通过，可以选择最后一步的审核环节。



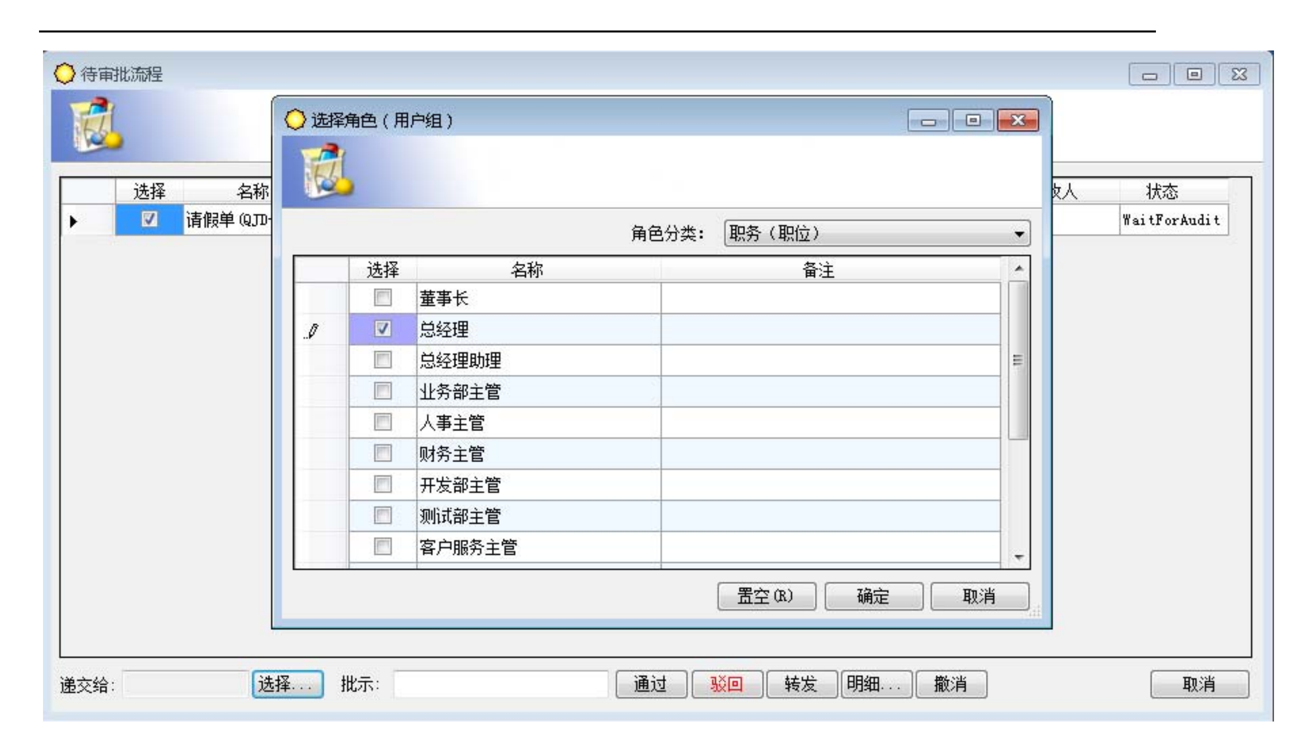


图22-12 进行审批

这里是递交审批的效果图



图22-13 递交审批

我们再以最后一个审批环节的帐号来登录系统



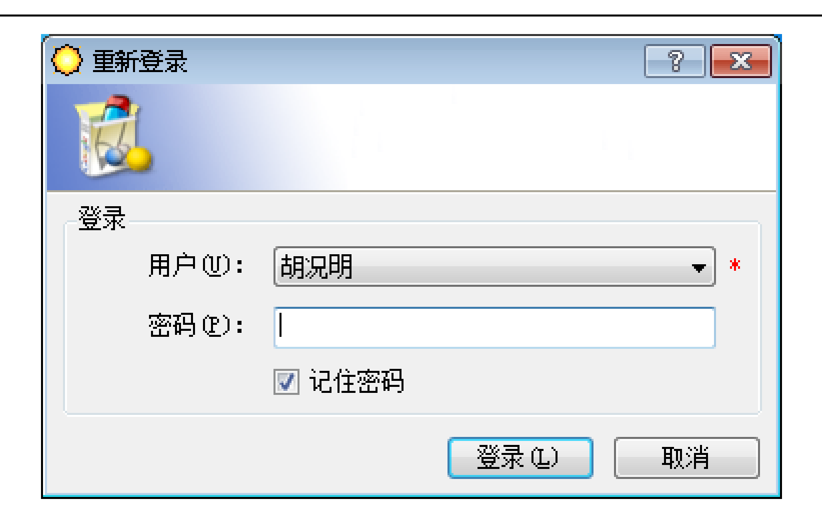


图22-14审批流程登录

我们可以看到前面2步的审批详细情况，我们这步审核通过后，我们事先假设的3步自由审流程也全部完成了。



图22-15 进行审批

自由审批流组件，有3个关键的功能。

1：当前需要审批的数据列表获取方法（可以按角色、按帐号兼容的）



2：查看审批流程历史记录（都谁什么时候审批的，有啥批示）

3：充当流程监控功能（都有哪些数据在待审核状态）

核心功能由2个组件组成

1：进入审批流程组件。

2：审批流程组件。

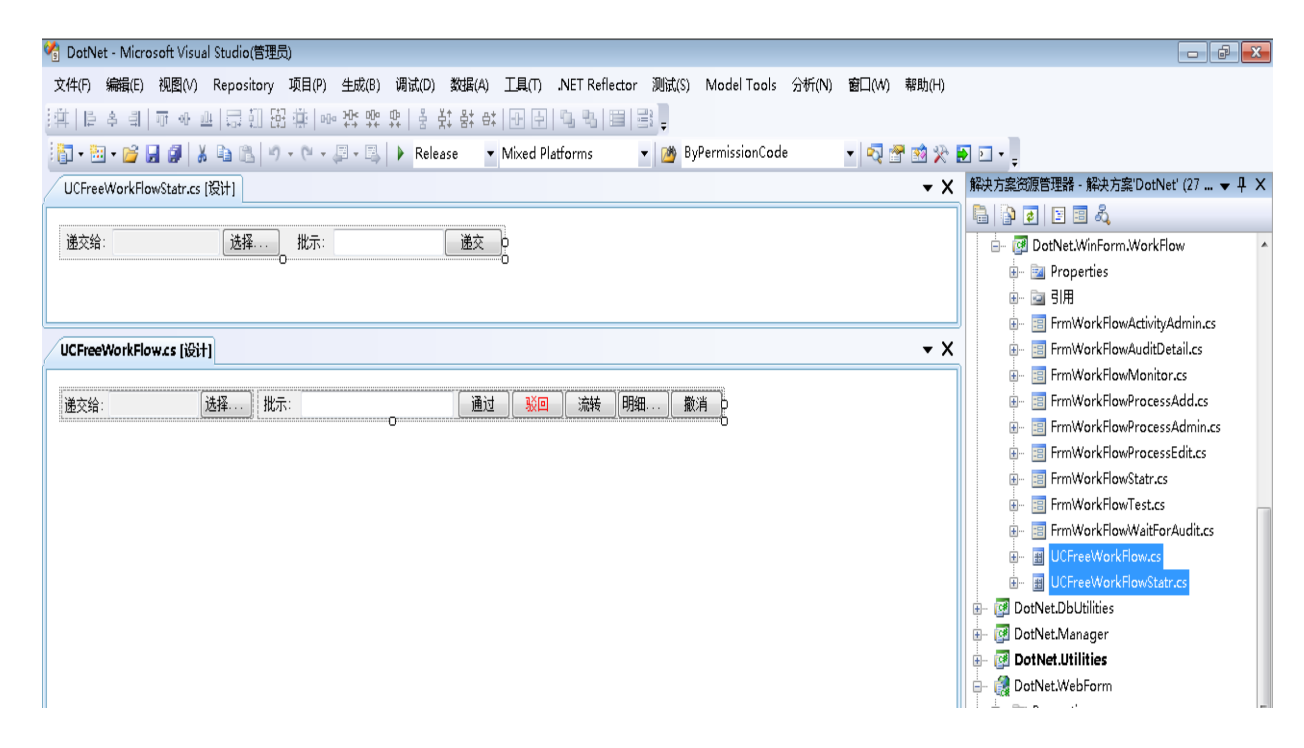


图22-16 审批流程组件

自由审批流程组件虽然从严格意义上来讲不能等同于工作流组件，甚至也不是正规的流程审核组件，但是还是很符合实际的工作需求，因为不需要事先定义好严格的流程步骤，也不需要定义严格的角色以及各种审批流程等，只要有最基础的用户数据，角色配置就可以跑起来了，相对而言是要求的配置最低最灵活多变的审批流程，更适合国内的中小企业的审核流程，特别是小公司的审核流程。

第24章　自动行政审批流程组件及流程节点的定义

审批流程组件的优点是请假流程等单据的审核过程不是固定的，是可以按客户的实际工作需要灵活配置的，是非固化的灵活的商业逻辑

程序员请假

递交请假、进入审核流程

开发主管审核

人事主管审核

总经理审核

审核流程结束
请假单生效

自动行政审批流程组件的审批流程节点定义。民间称为工作流组件官方称为审批流程组件。上一讲中，请假单的流程步骤是由人为自由流转的，虽然自由流转有很多好处但是并没有严格的流程步骤限制，想给谁审核就怎么审核，自由审核流程组件主要存在以下2个重要的缺点：

1：流程步骤没有规范的定义，可以理解为乱来的，特别是公司规模大了，人多了，那就更容易乱套了。

2：流程没有重复利用的价值，属于公司的管理办法没能被固化、流程化，若有固定的流程可以不断优化，那公司的管理也上档次了，更规范了。

为了解决以上2个问题，先来看看如何在系统里预先定义一个简易的审批流程，先

把审批流程定义部分的文章整理如下：

1：首先我们用超级管理员的身份登录系统

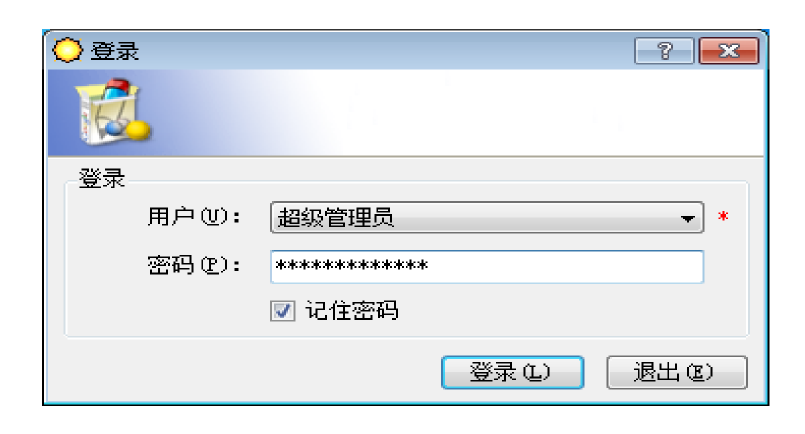


图23-01审批流程登录

由于超级管理员有系统的任何权限，所以可以操作系统里的任何菜单



图23-02审批流程定义管理

2：我们进入审批流程定义管理菜单，来定义我们需要的审批流程

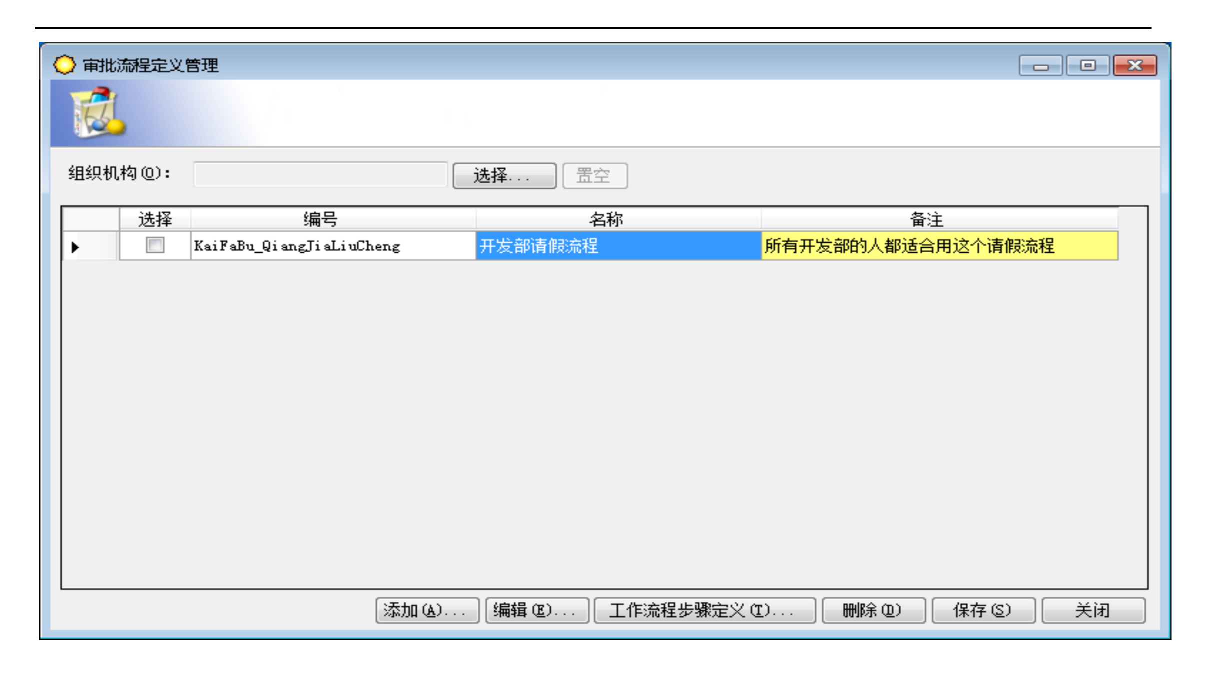


图23-03审批流程定义管理

3：审批流程，大多是按部门来组织的，例如哪个部门的哪个审批流程等，所以也有按部门进行过滤查询的功能，我们来定义具体的审批流程

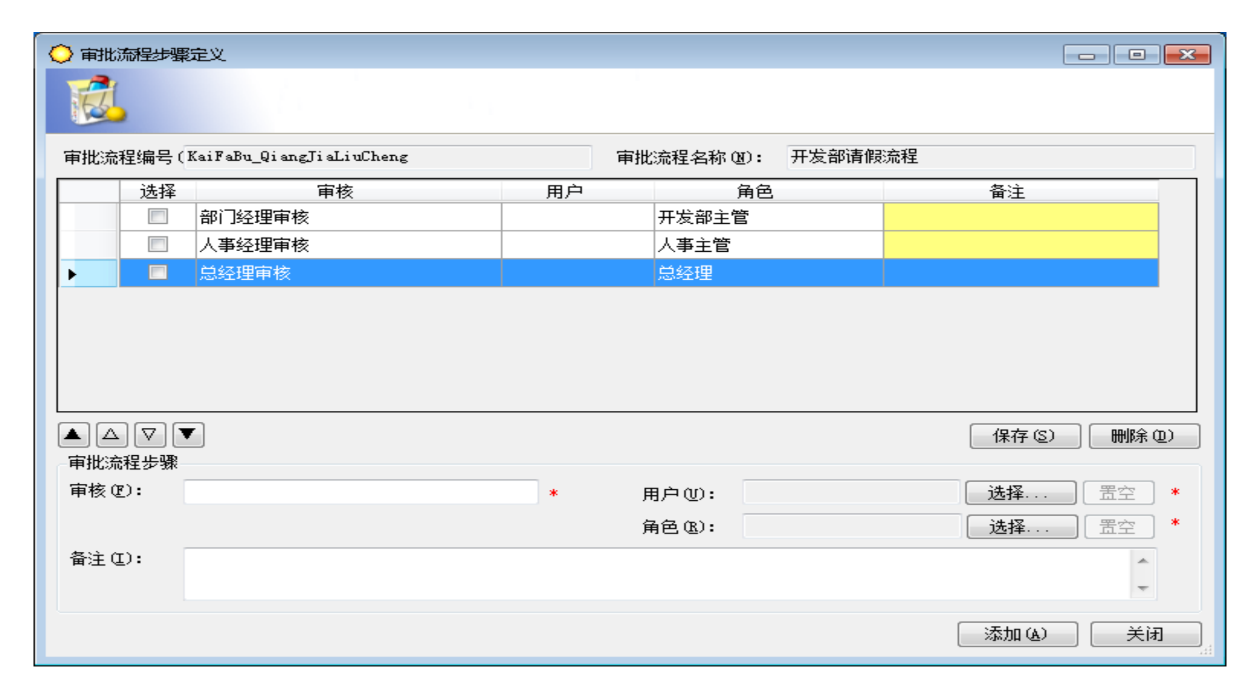


图23-04定义审批流程

我们定义好了3个流程审核步骤后，接下来应用程序里就可以按以上审核步骤进行自动流转进行审批操作了，图形化的审批流程组件会后续发布，敬请等待.

相应的数据库设计如下：

1：工作流定义表：定义我们总共有哪些审批流程？

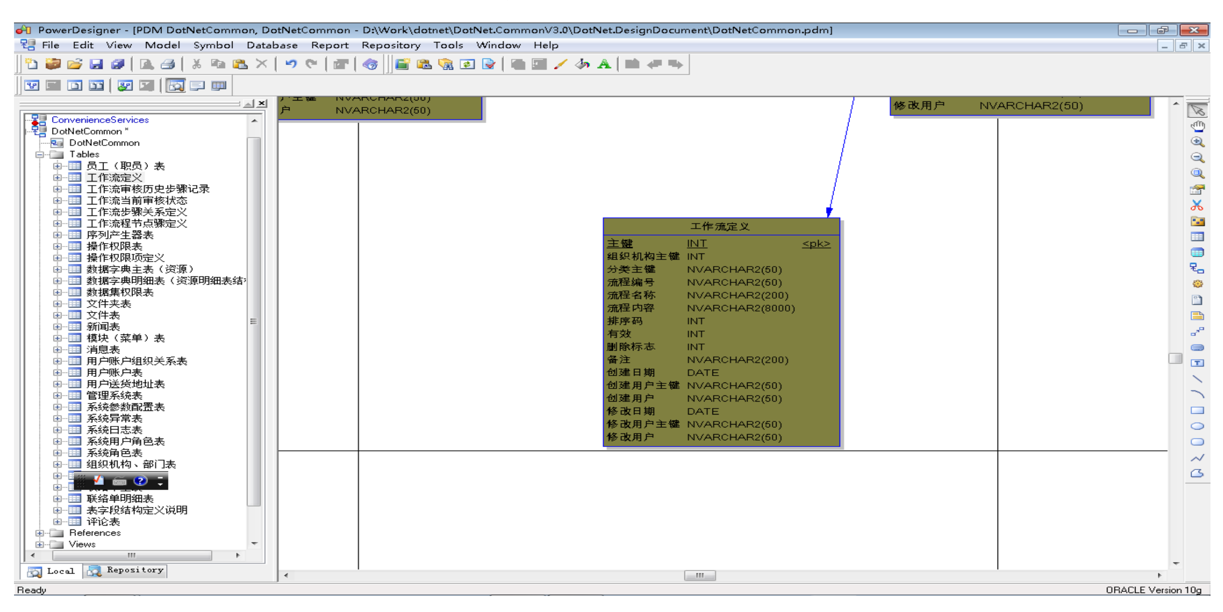


图23-05定义审批流程类型

2：工作流程节点定义表：定义我们的审核流程节点都有哪些？

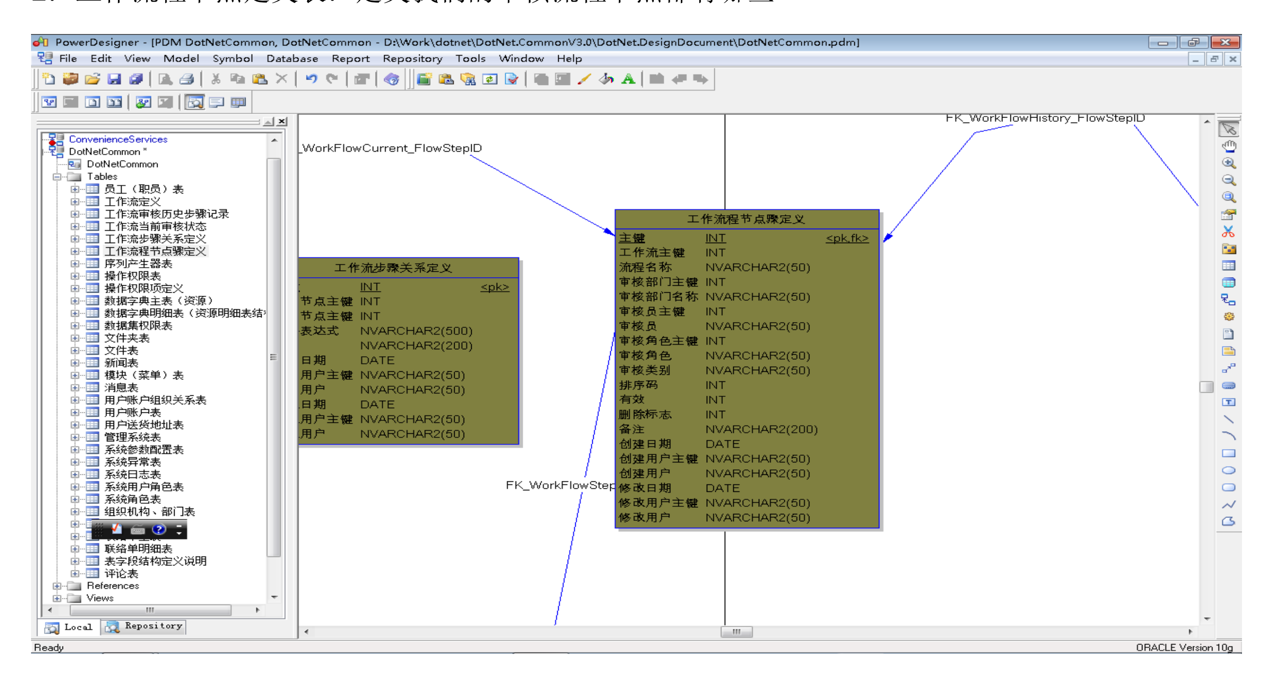


图23-06定义审批流程表

定义好审批流程后，我们用相应的账号登录系统后，观察效果。首先我们以超级管理员登录



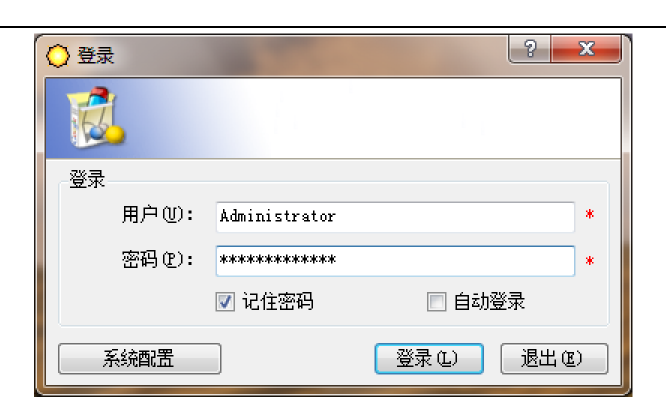


图23-07审批流程登录

我们先启动一个自由行政审批流程



图23-08启动审批流程

假设我们按了自动审批流程控件的提交按钮



图23-09审批申请

用吉日嘎拉的系统帐号登录，进行第一步审核

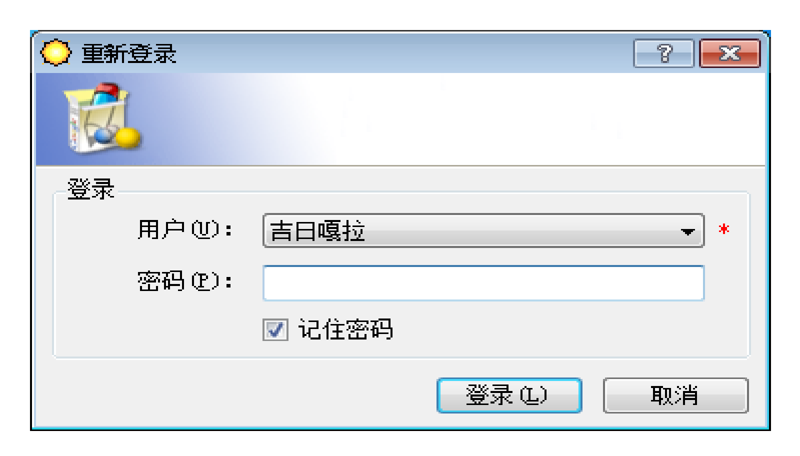


图23-10审批流程登录

吉日嘎拉账号登录系统后，会有自动的审核流程的即时通讯提醒信息，我们进入待审核流程页面

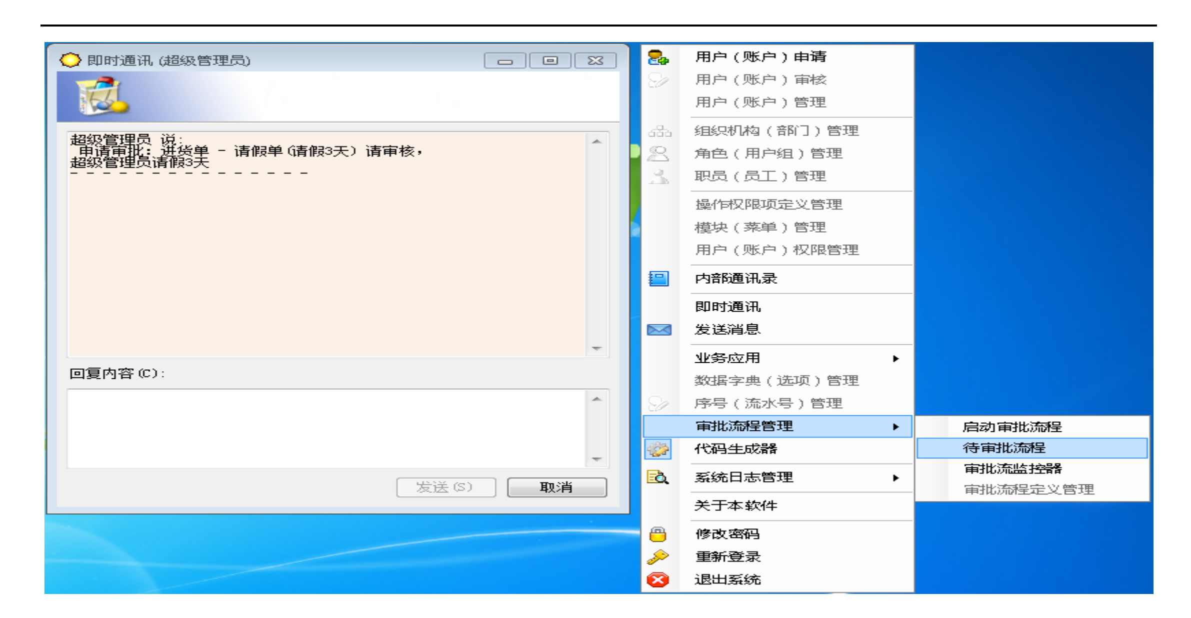


图23-11查看待审批流程

我们可以按红色圈起来的，自动审核流程中的通过按钮。



图23-12批准申请

以此我们可以用涂志刚账号，进行第2步审核，也会即时通讯提醒信息收到，也可以进行审核，页面省略

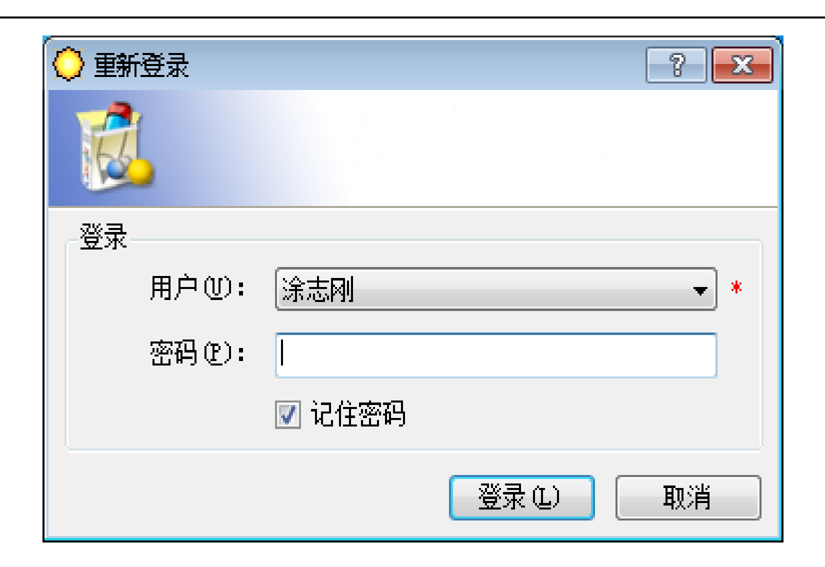


图23-13审批流程登录

再次我们可以用胡况明账号，进行第3步审核，也会有即时通讯提醒信息，也可以进行审核，页面省略

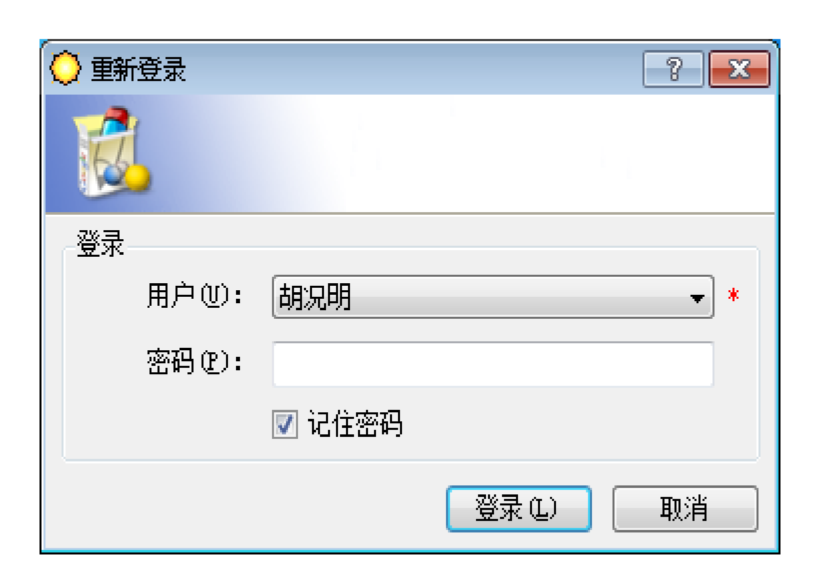


图23-14审批流程登录

最后我们可以在审核流程监控器中可以看到这个请假单的详细审核步骤的信息，一个完整的自动行政审核流程就可以顺利进行完毕了。



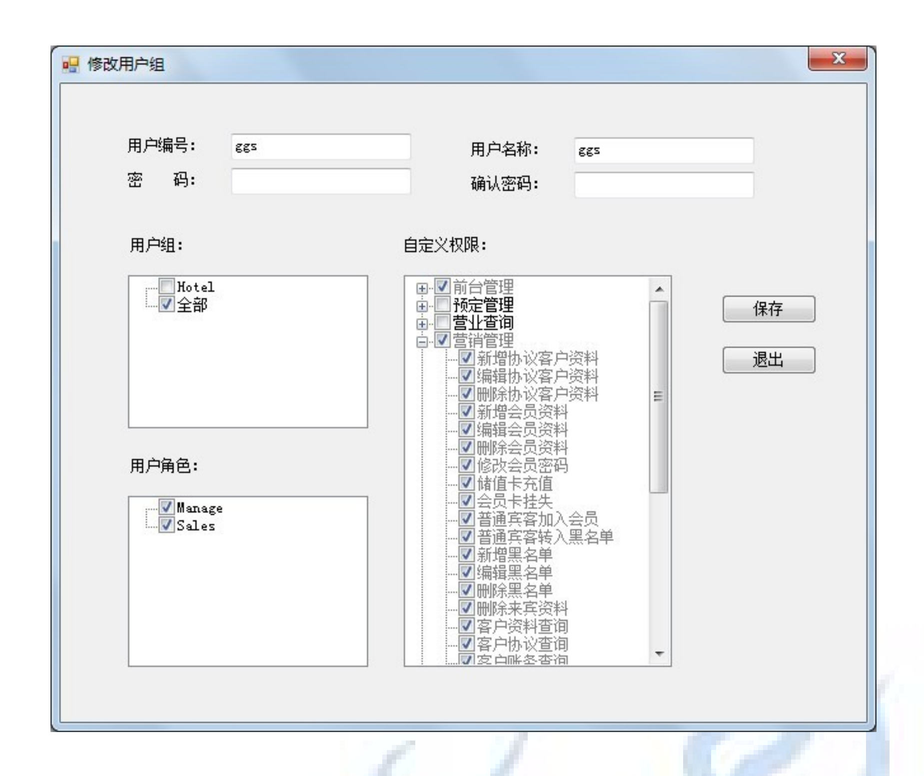
图23-15审批全部完成



第25章　系统权限批量设置

平日里空了就收集一些权限设置页面或者参考别人是怎么实现权限配置的，其实收集多了也很有意思的，体会体会别人的设计、别人的意图，也是件其乐无穷的事情。

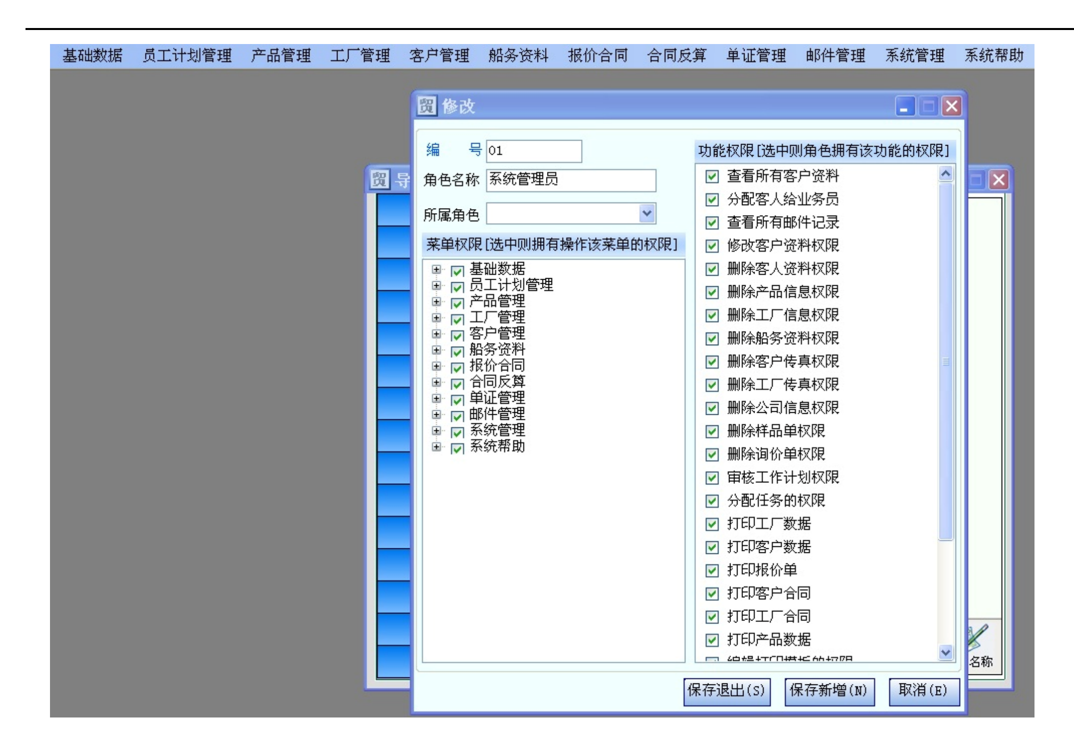
下面我们看看，其中某一个人的权限设置页面如下图：



虽然页面设计布局一般，但是此权限配置程序用起来还是会蛮顺手的，只是有些觉得用户组、角色有些重复，不知道其中有什么奥秘？还有值得表扬的就是操作权限设置也不错，但是不知道菜单权限在哪里设置了？难道全部通过操作权限来实现？进行硬编码控制菜单的访问权限？

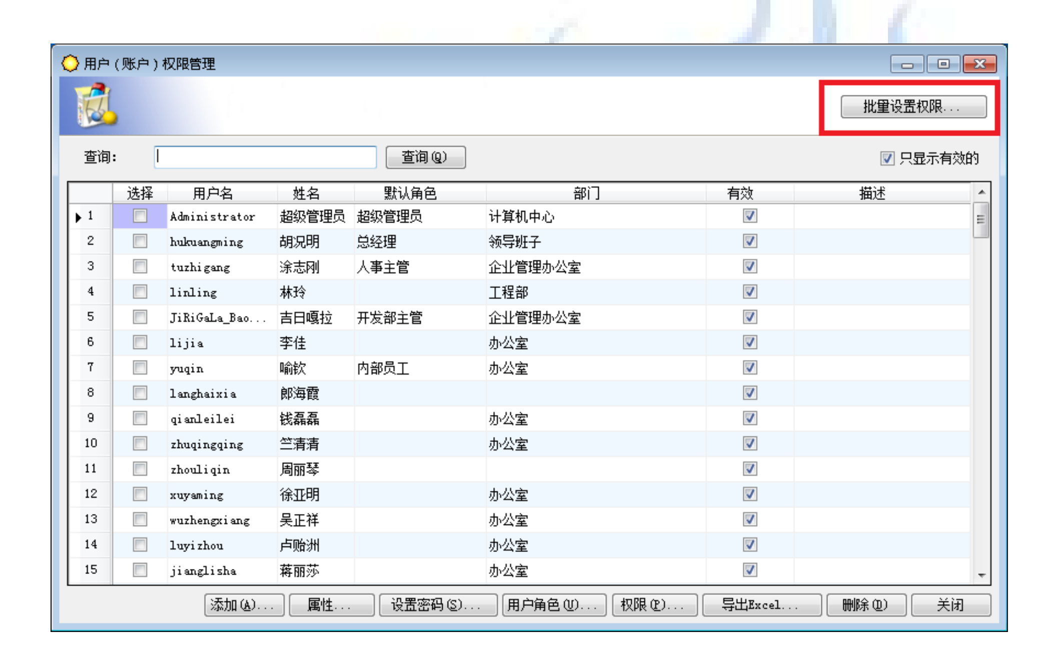
还有感觉这个页面更像是新增页面，而不像是管理配置页面，因为管理配置页面不是每次都修改密码的吧？若我没理解错，这个应该是新增用户时的页面并同时实现了配置权限的功能，设计还是有些合理的。

我们再看看另外一个人的权限配置的页面如下图：

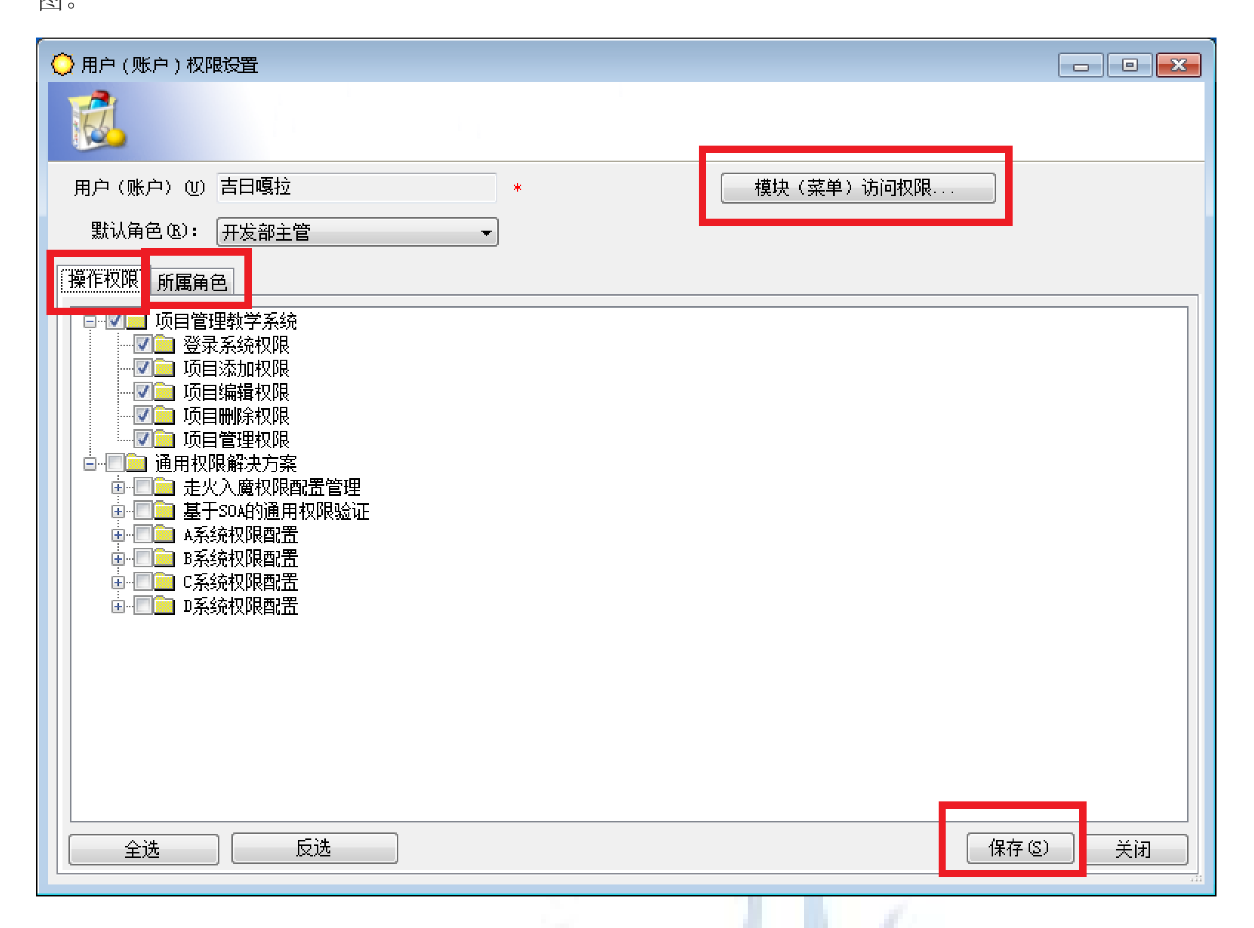


其实这个权限设计开发人员的功底还是蛮深的，菜单访问权限、操作权限都可以进行灵活设置，只是没太了解一个角色还有所属角色，不是非常好理解，也不知道为什么要这么设计了？

再看看通用权限管理系统组件里的权限配置页面效果如下图：



下图是点选了某个用户后点“权限”按钮后弹出的权限配置页面效果图。



上面的页面，虽然从功能上讲操作权限、归属角色、模块菜单访问权限等等都可以进行设置，但是经常用会发现有几个问题存在：

1：每次设置权限都要切换好几个页面，然后还要点好几个按钮，用起来稍微麻烦一些、有点儿折腾、而且功能不直观。

2：不好进行批量权限设置，某个人要设置权限非需要点一下权限，然后设置权限保存好后再设置另外一个人的，操作步骤多，使用的人容易眼花缭乱。

我们按 2 个国家之间的战争做个比喻：

同样 2 个国家都在用枪炮互相打仗，但是一个国家相对落后都是用的是步枪，步枪虽然也有很多优点例如打得远、省子弹啥的；



但是另外一个国家相对发达全部配备了机关枪，机关枪虽然也有缺点，例如打得不够远、子弹容易到处乱飞、不经济；

你若在战场上，希望加入用的是步枪的国家、还是用机关枪的国家？ 我想大多数不想送死还是会选择用机关枪的，火力威猛更容易胜利好保住小命。

下面这个改进型的批量设置权限功能，属于火力威猛的机关枪类型，操作连贯、又直观、又不用关闭弹出窗口，也不用切换功能模块，也不用每次按保存按钮，适合批量设置权限，一天估计能把几百几千人的权限设置好，有点儿属于全自动机关枪的感觉。

若是为了打猎那步枪就很合适了，若是战争还是得用机关枪好，通用权限管理系统就是为了给快速开发的大型软件项目准备的，让你用得爽，甚至半天就把用户帐号权限等统统都配置好，在这方面就不要耗费什么精力了，省下的时间安心该干啥就干啥就可以了，省下时间可以泡妞去了，改善一下生活质量，提高一下生活情调。





其实，很多人都能实现功能，但是未必深入就体会，天天用某个软件，只有天天用用了10 年 8 年，会有非常深刻的体会，就能感觉出哪个软件是好用的，哪个软件是不好用的，应该怎么改进才是好用的。

夏利车也是车奔驰车也是车他们功能都差不多，但是差距往往体现在很多细节上，通用权限管理系统也是一样的，很多人并不是不能实现这些功能，但是未必做得这么精、这么深入、为客户想得这么体贴，为客户日常操作着想 N 多问题，并不只是考虑了功能、性能等开发人员最关注的问题，好就好在很多细节处理实现上。



第26章　实现基于组织机构的数据集权限

绝大部分内部管理系统都需要实现各种复杂的操作权限限制，而不是单纯的把所有数据都共享出来，特别当涉及到一些机密的商业数据或者财务数据时，都需要进行严格管控，当遇到类似的需求时往往需要一个可以灵活配置数据集权限的管理工具来实现客户的实际需要，同时需要有相应的公开的 API 函数可以过滤数据，提高我们的建设内部管理系统的工作效率。

当然实现一个这样的权限管理系统，也不是很难的事情，可能 1 个人花费几个个月时间也可以做好了，但是做好的系统由于是匆忙赶工的，很可能会导致以下几个问题。

1：考虑不够充分，没有扩展性，想修改个啥功能很难。

2：由于没有足够的开发时间，可能代码质量，数据库设计等都没严格检查，将来维护起来也闹心一些。

3：虽然现在是有精力做一个，但是没精力一直维护下去，因为手上需要做会越来越多，根本顾不来这个了。

4：没经过几年的实战，程序经不起折腾，往往遇到一个问题或用到其他某个项目里，需要修改的地方很多。

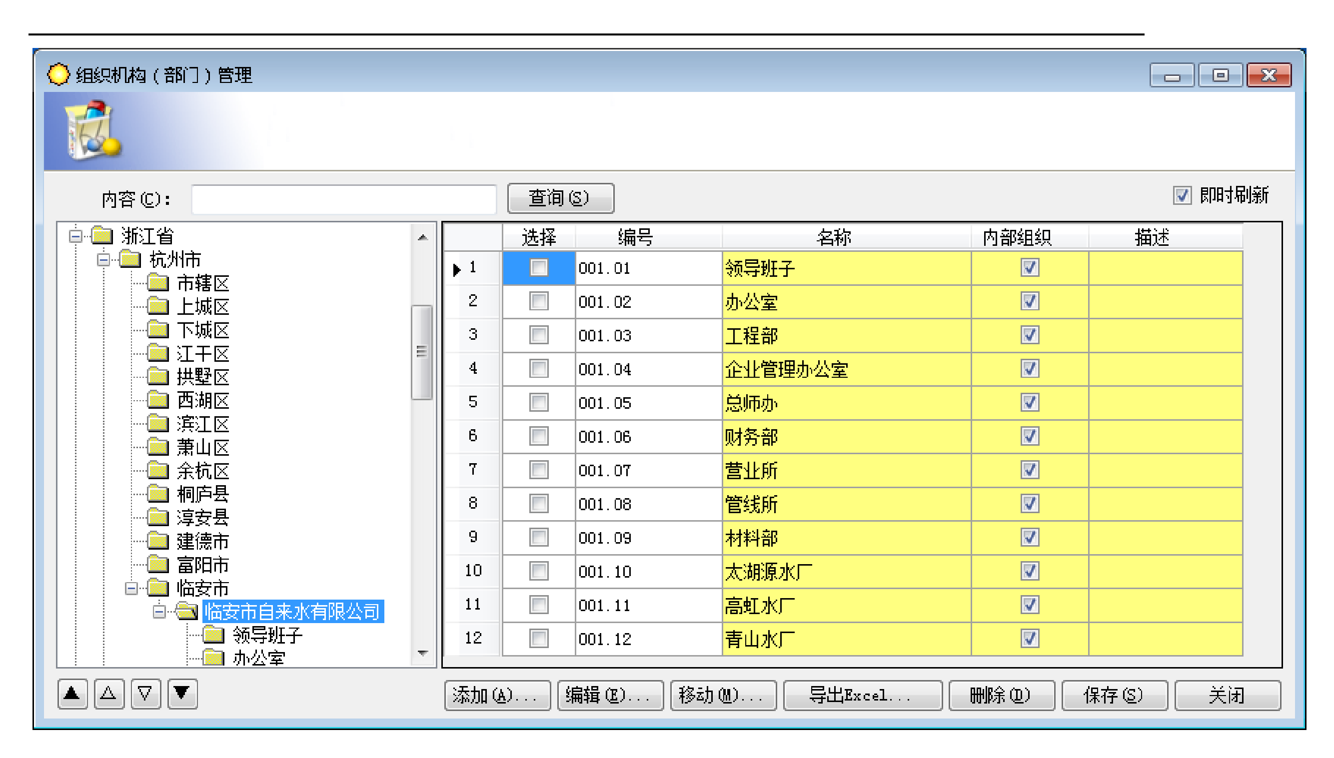
5：由于程序是从慢赶工的，很多东西都没能确定下来，导致不断的修正，影响了其之上开发的应用，三天两头变。

6：可能只考虑了功能，没考虑客户的易用性，操作起来不方便。

7：虽然程序是好用了，但是没配套的文档，没数据库设计文档，没操作手册，只是一个半成品。

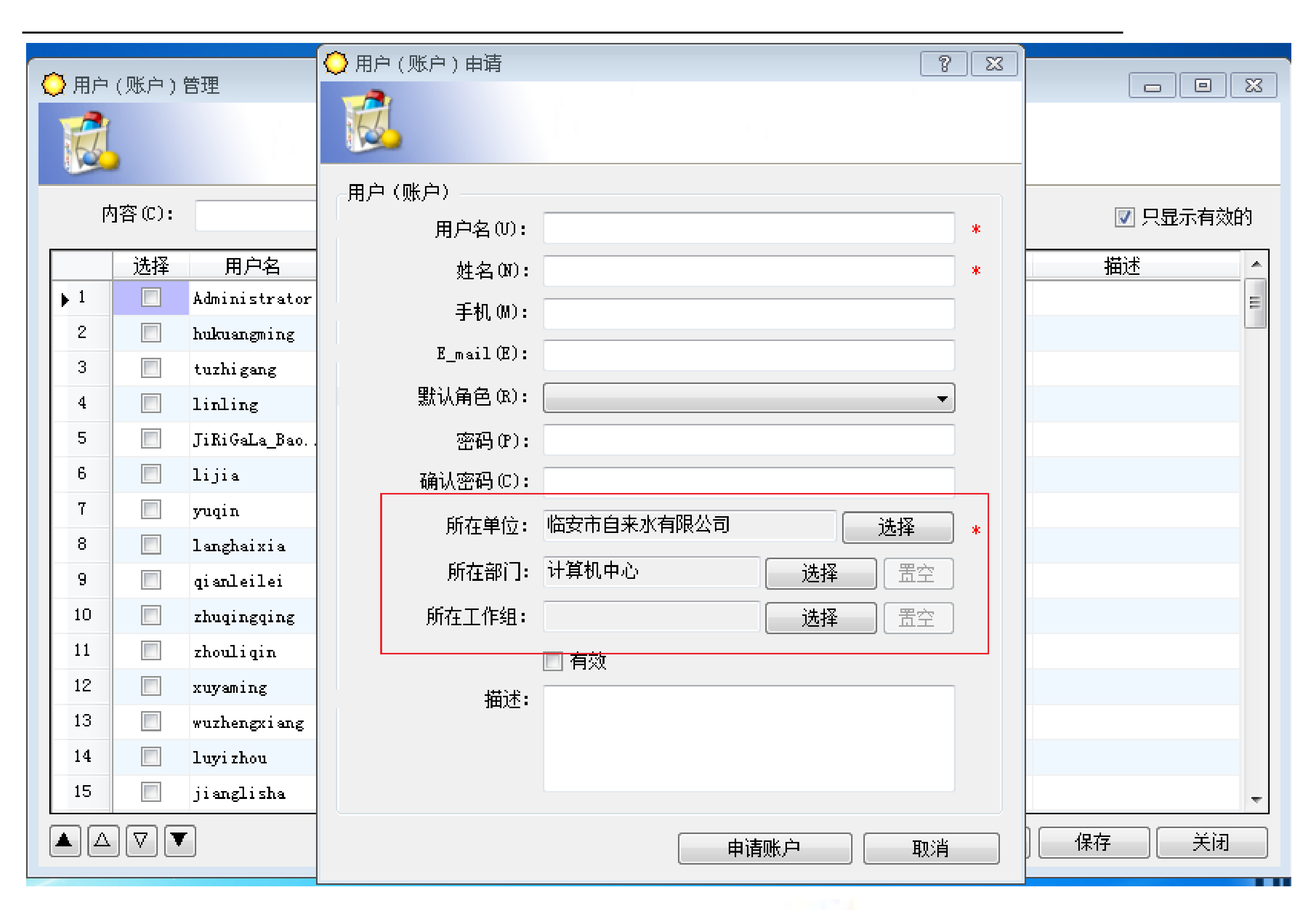
接下来我们讲解如何现基于组织机构的数据集权限系统

01：首先我们需要实现公司的组织机构管理，公司从整体上划分了哪些职能部门等。

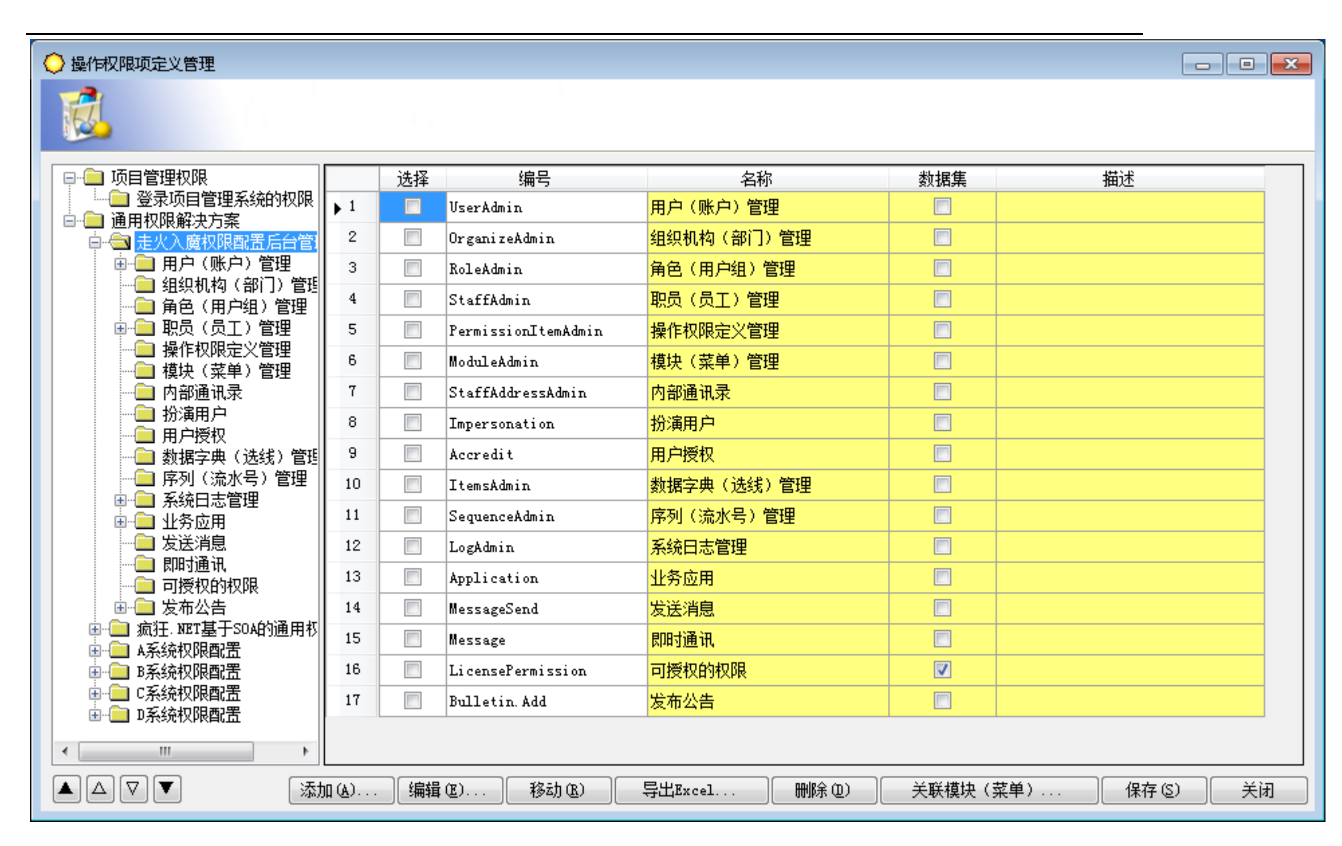


02：其次是我们需要管理用户，这些用户又是属于哪个公司、哪个部门、哪个工作组等，这些是会登录到系统的用户帐号，我们先不去管角色、用户组什么的，那个另外谈，相对简单的东西，我们先把整体思路理顺好。

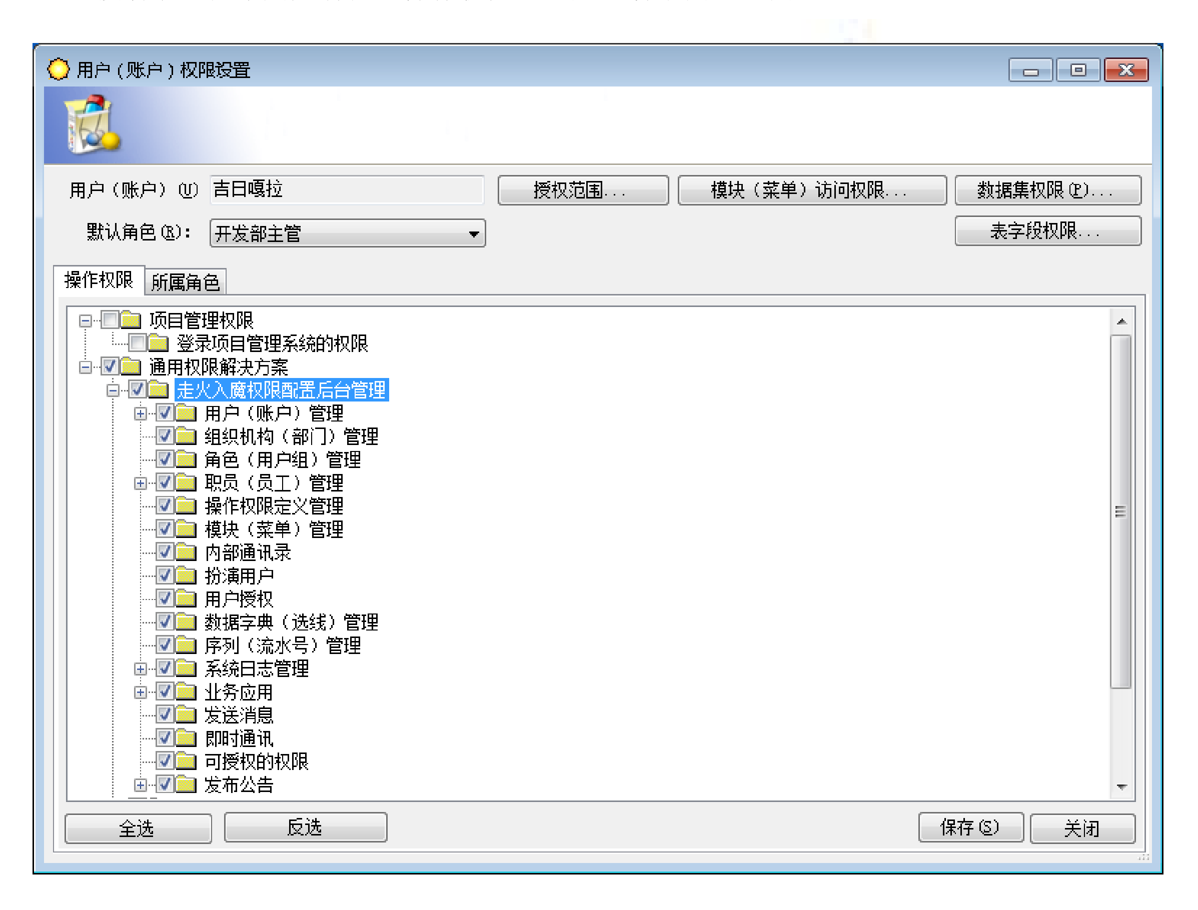




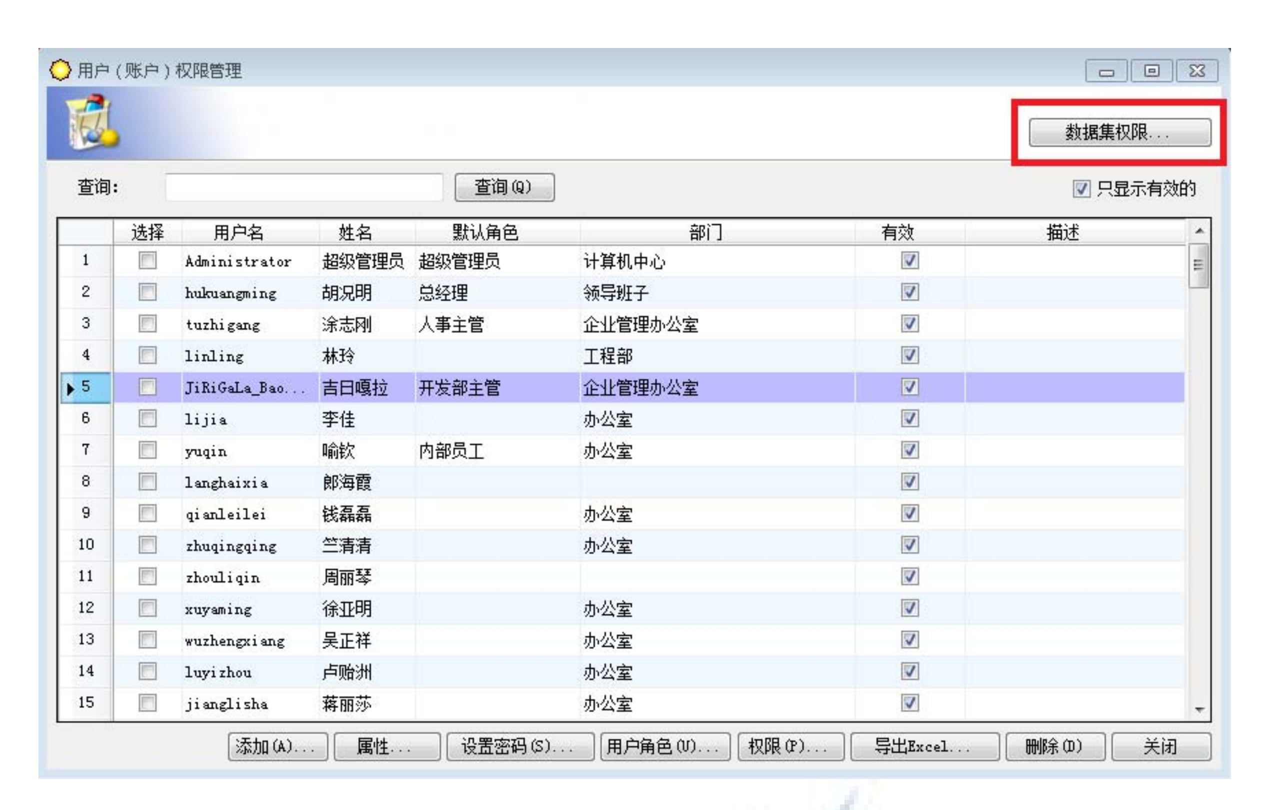
03：我们系统里需要定义好操作权限，系统里到底需要管理控制哪些权限？这些权限都可以灵活设置，那就比较理想了，往往很多系统这一步控制得不是很好，定义权限不够灵活，不好扩展，不好整合。



04：接着就是哪个用户有哪些操作权限的问题，看下面的画面。

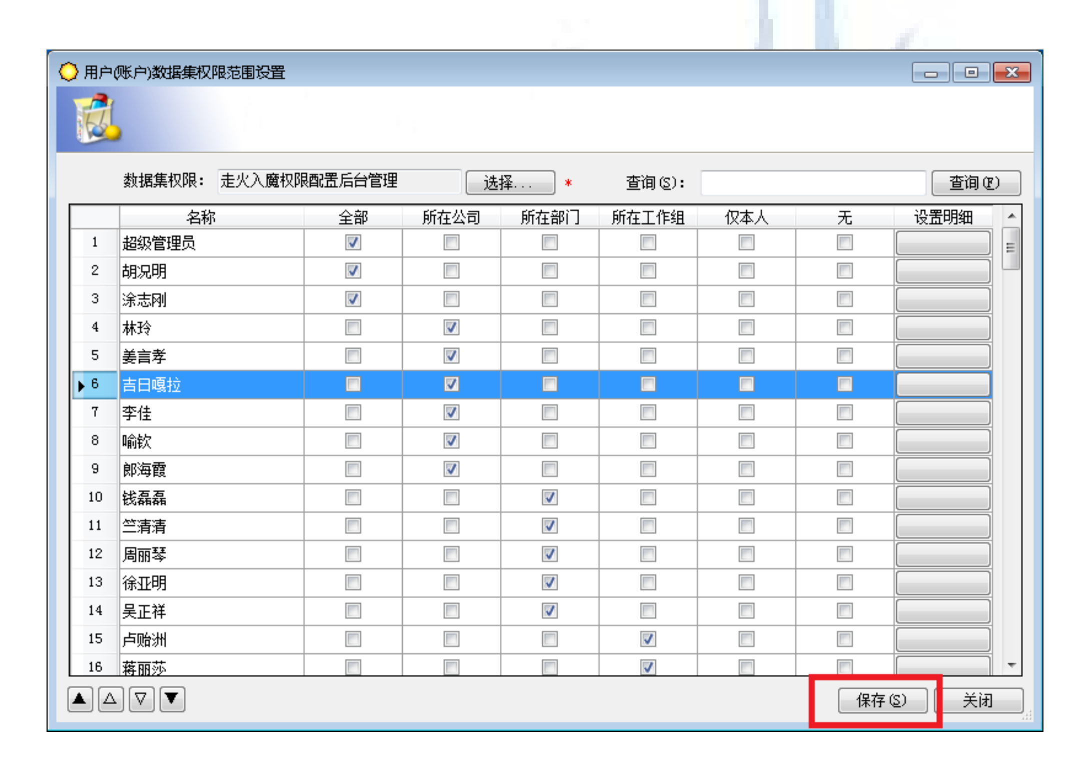


05：有后台权限管理的权限，但是不是对所有的数据有管理权限，只对一部分数据有管理权限？那我们怎么来设置权限管理条件呢？看下图



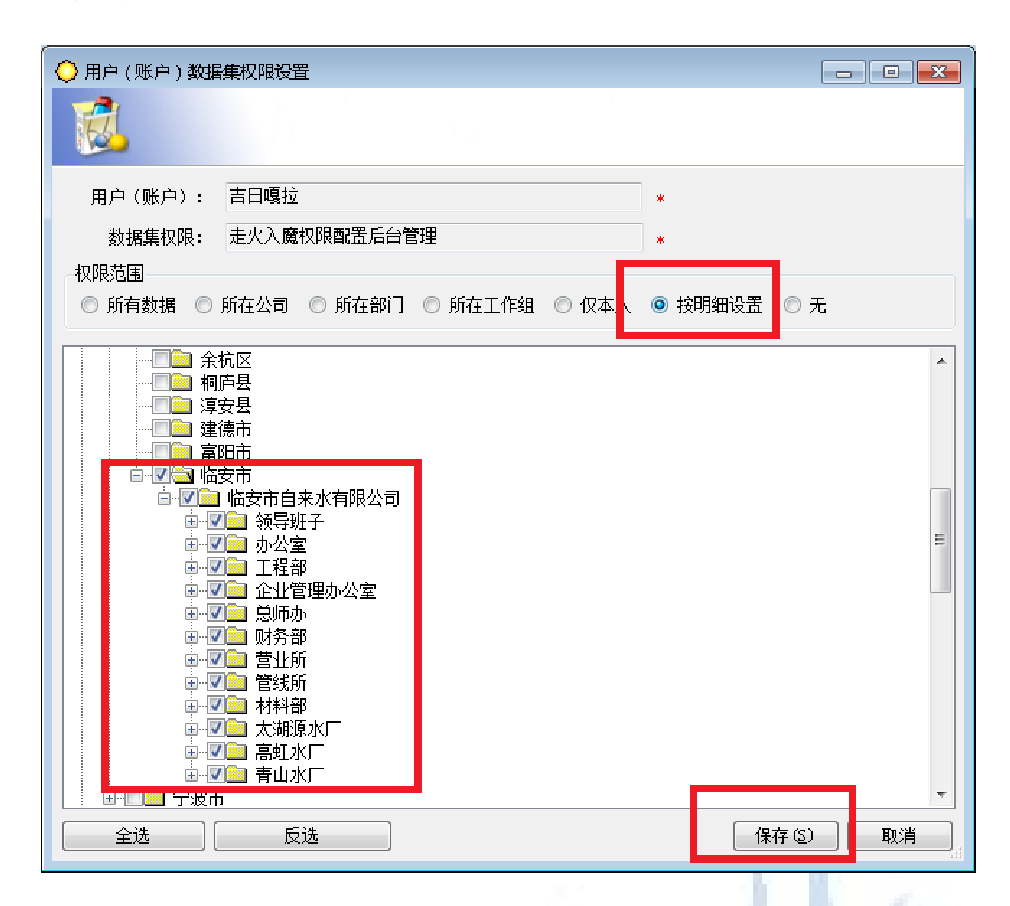
	选择	用户名	姓名	默认角色	部门	有效	描述
1	☐	Administrator	超级管理员	超级管理员	计算机中心	☑	
2	☐	hukuangming	胡况明	总经理	领导班子	☑	
3	☐	tuzhigang	涂志刚	人事主管	企业管理办公室	☑	
4	☐	linling	林玲		工程部	☑	
5	☐	JiRiGaLa_Bao...	吉日嘎拉	开发部主管	企业管理办公室	☑	
6	☐	lijia	李佳		办公室	☑	
7	☐	yuqin	喻欽	内部员工	办公室	☑	
8	☐	langhaixia	郎海霞			☑	
9	☐	qianleilei	钱磊磊		办公室	☑	
10	☐	zhuqingqing	竺清清		办公室	☑	
11	☐	zhouliqin	周丽琴			☑	
12	☐	xuyaming	徐亚明		办公室	☑	
13	☐	wuzhengxiang	吴正祥		办公室	☑	
14	☐	luyizhou	卢贻洲		办公室	☑	
15	☐	jianglisha	蒋丽莎		办公室	☑	

06：我们看看快速设置数据库权限的效果图

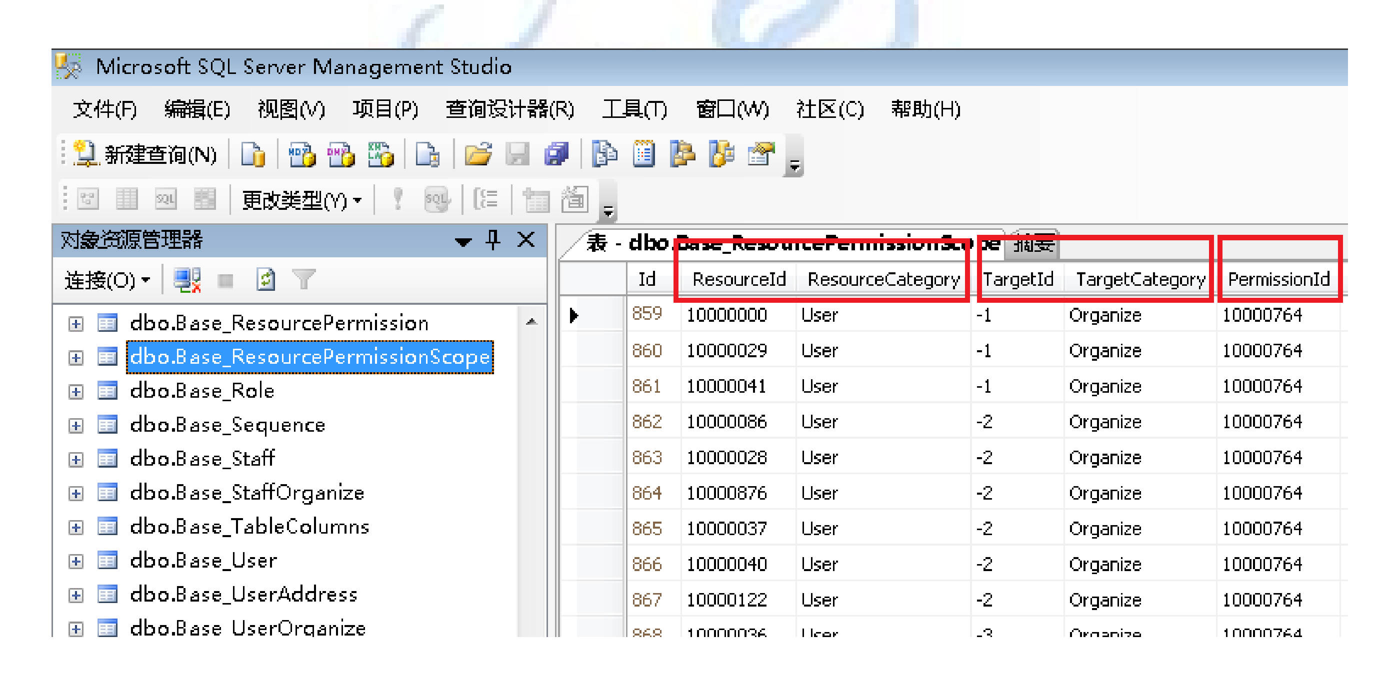


	名称	全部	所在公司	所在部门	所在工作组	仅本人	无	设置明细
1	超级管理员	☑	☐	☐	☐	☐	☐	
2	胡况明	☑	☐	☐	☐	☐	☐	
3	涂志刚	☑	☐	☐	☐	☐	☐	
4	林玲	☐	☑	☐	☐	☐	☐	
5	姜言孝	☐	☑	☐	☐	☐	☐	
6	吉日嘎拉	☐	☑	☐	☐	☐	☐	
7	李佳	☐	☑	☐	☐	☐	☐	
8	喻欽	☐	☑	☐	☐	☐	☐	
9	郎海霞	☐	☑	☐	☐	☐	☐	
10	钱磊磊	☐	☐	☑	☐	☐	☐	
11	竺清清	☐	☐	☑	☐	☐	☐	
12	周丽琴	☐	☐	☑	☐	☐	☐	
13	徐亚明	☐	☐	☑	☐	☐	☐	
14	吴正祥	☐	☐	☑	☐	☐	☐	
15	卢贻洲	☐	☐	☐	☑	☐	☐	
16	蒋丽莎	☐	☐	☐	☑	☐	☐	

07：若以上设置，还不能满足要求，那就可以进行更详细的设置，点明细按钮会出来下面的设置界面

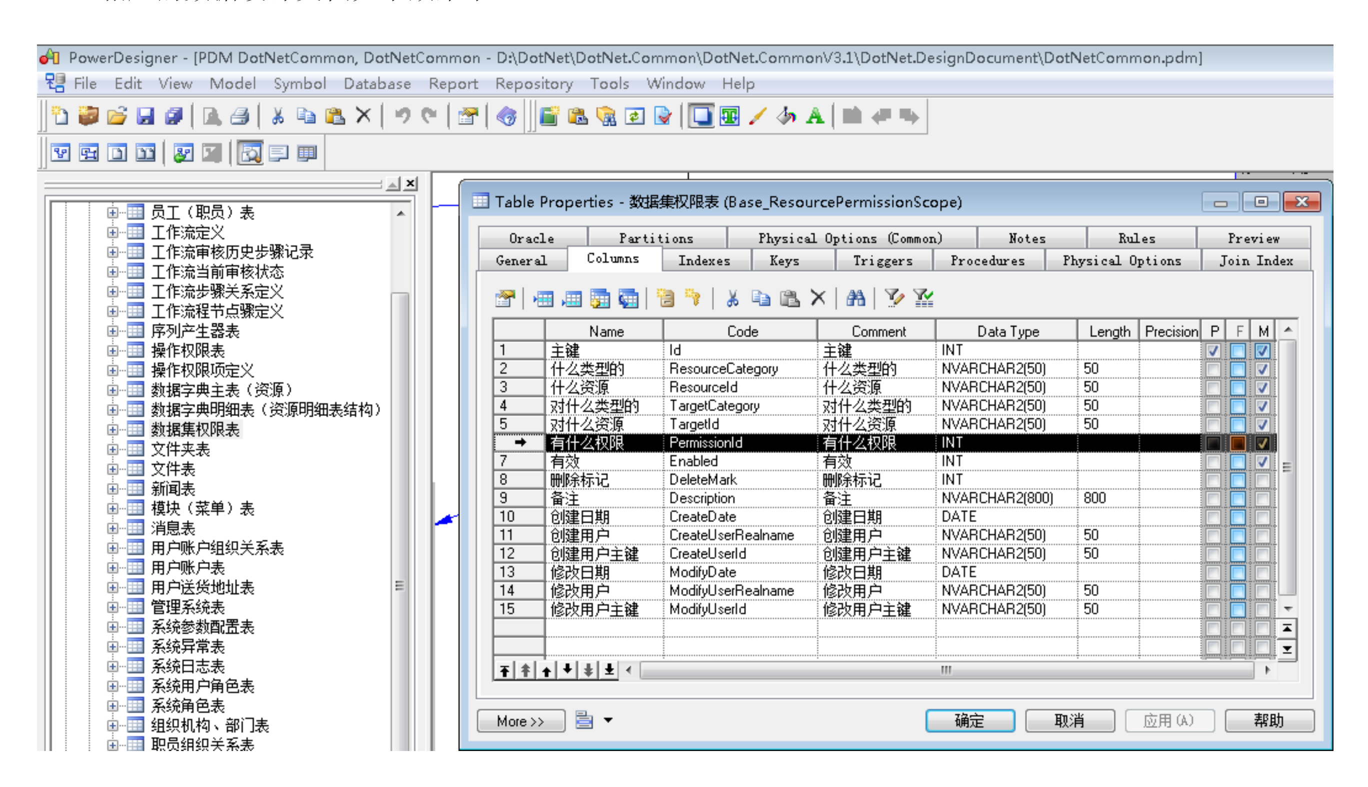


08：我们在看看相应的数据库里的数据保存效果



Id	ResourceId	ResourceCategory	TargetId	TargetCategory	PermissionId
859	10000000	User	-1	Organize	10000764
860	10000029	User	-1	Organize	10000764
861	10000041	User	-1	Organize	10000764
862	10000086	User	-2	Organize	10000764
863	10000028	User	-2	Organize	10000764
864	10000876	User	-2	Organize	10000764
865	10000037	User	-2	Organize	10000764
866	10000040	User	-2	Organize	10000764
867	10000122	User	-2	Organize	10000764

09：相应的数据设计文档如下效果下





第27章 C/S 组件模式运行用例

其实大多情况下，都不想了解细节，只是能用就可以了，例如买了一个菜刀，只是为了做饭时用用，而并不想关心制作菜单过程中的细节、甚至炼钢的细节等等，通用权限管理组件也是一样的道理，只是拿过来能用、好用就可以了。

下面的程序是以 C/S 插件模式运行的效果，做了一个最简单的主程序，然后在主程序里调用各权限组件的实际运行效果。

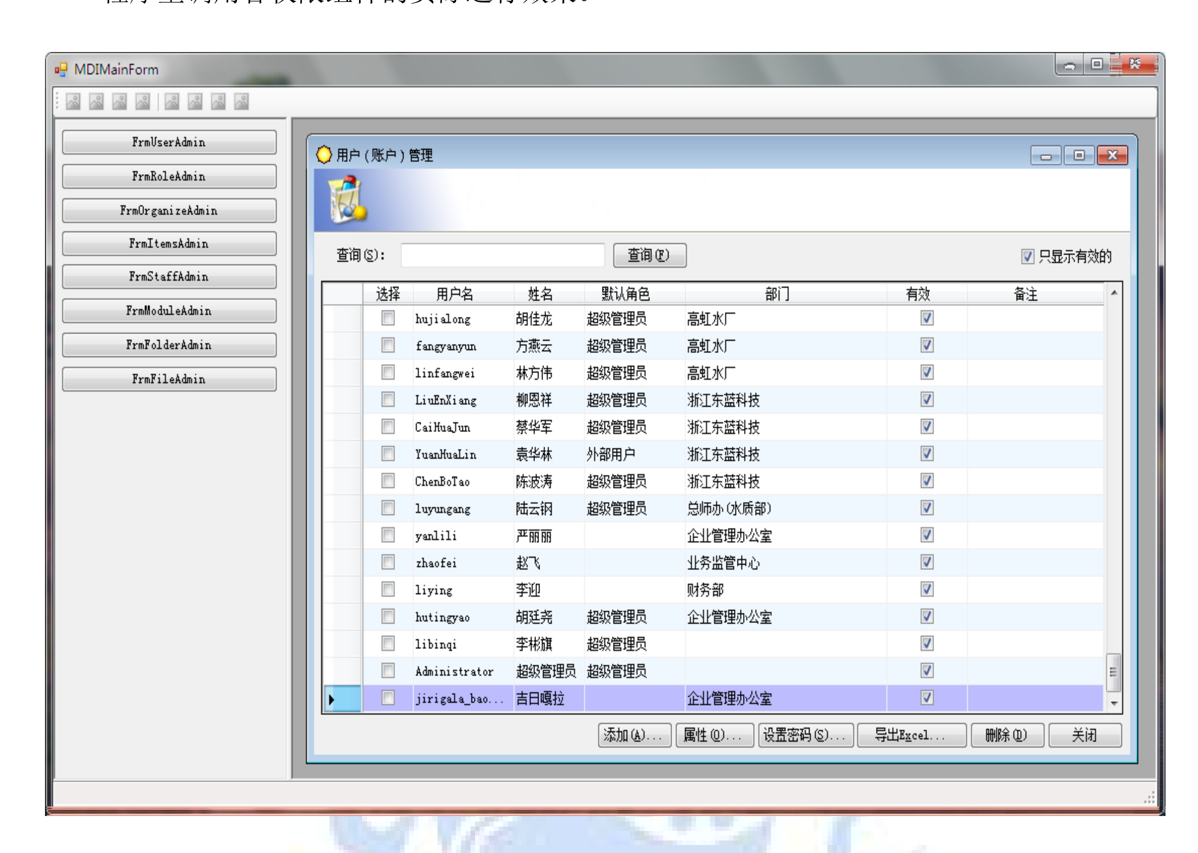


图 24-01 限组件的实际运行效果图

走火入魔通用权限管理组件中，有 2 个主程序可以选择性的配置，SDIMainForm 为以管理配置工具的运行效果，MDIMainForm 为以插件模式运行的效果，可以在 Config.xml 里配置，到底选用哪个主程序，或者选用哪种登录窗体等。

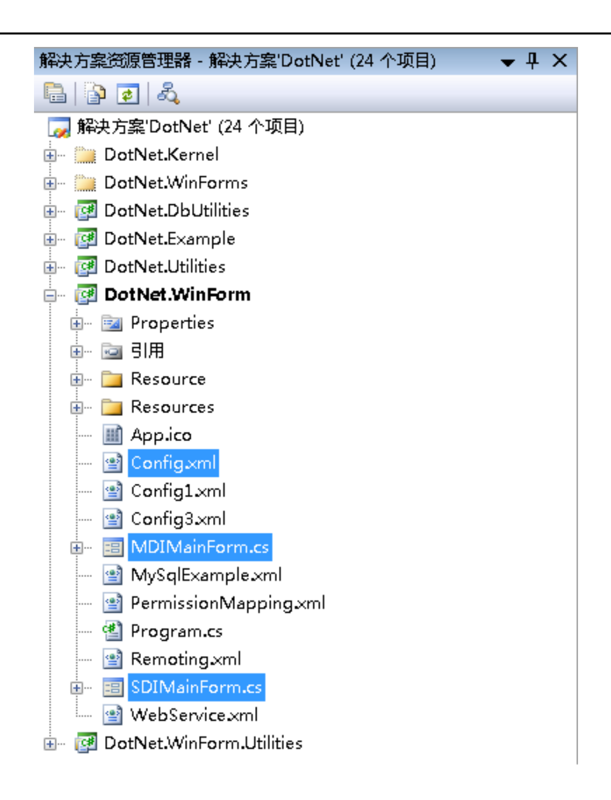


图 24-02 选用登录窗体

各个功能窗体的源码如下图，子功能模块切割得很细、分工很明确。

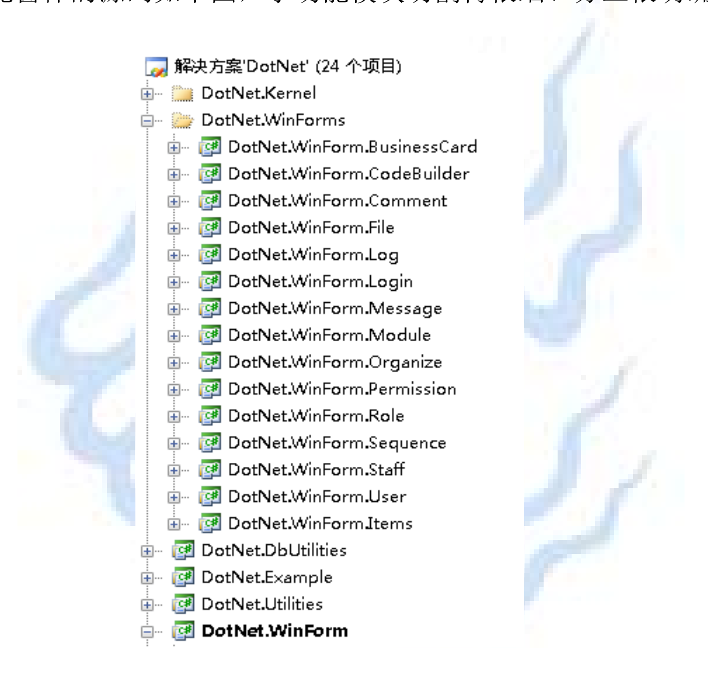


图 24-03 功能窗体源码

下图是项目引用方式的效果图。



图 24-04 项目引用方式的效果



第28章 B/S 组件模式运行用例



第29章 多语言支持

虽然平时很少接触老外的项目、也很少碰到老外，但往往赚大钱的人是经常跟老外做买卖的人居多，他们大多需要网站是全英文的，往往上手有一个成熟的中文的网站或者软件，例如成熟的 B2C 网上购物系统，但是没办法也没精力再做一个全英文的系统维护，只能让手上的项目既能支持中文也能支持多语言版本，还是维护一套程序比较可行一些，同时维护改进好几套程序，还是有点儿难度，工作量也会相对加大。

碰到什么生意就能做什么生意，客户需要个什么语言的包就给做个什么语言包的，平时需要把准备工作做好，前些日子就遇到了这样的事情，B2C 网上购物的项目，需要给外贸公司做个给老外展示的平台、需要做成全英文的，做了相应的英文语言包，这个问题就得到解决了，平时只维护一套程序就可以了，客户负责把英文语言包维护好，也不用大改程序，直接文本文件修修改改就可以了，直接可以修改文本文件就可以的往往是效率最高、最简单、最高效、最经得起考验的做法。

多语言的实现参考如下图，这是语言包以 XML 文件的格式、直接存储了文本文件、需要几个国家的语言支持，只要有配套的包就可以了。



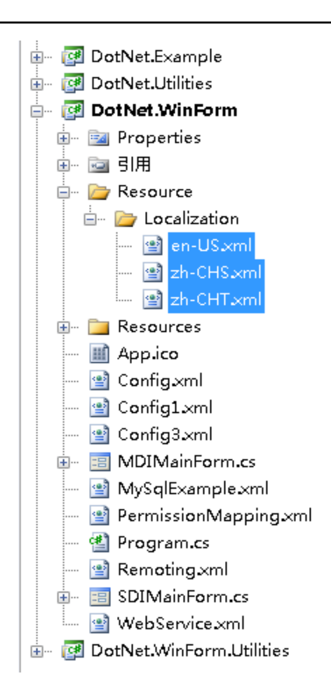


图 24-01 多语言的实现

这是语言包里的内容参考，一些常用词汇的定义。

en-US.xml

```xml
<?xml version="1.0" encoding="utf-8" ?>
<resources>
  <language>en-US</language>
  <displayName>American English</displayName>
  <version>1.0</version>
  <author>JiRiGaLa</author>
  <description>American English Language Package</description>
  <items>
    <!-- Global Language Resources Begin JiRiGaLa -->
    <item key="MSG0000">Notice Messsage</item>
    <item key="MSG0001">Unknown error occured. </item>
    <item key="MSG0002">Database connection exception. </item>
    <item key="MSG0003">Webservice connection exception. </item>
    <item key="MSG0004">No data modified. </item>
    <item key="MSG0005">Record did not found, may be deleted by other users. </item>
    <item key="MSG0006">Data has been modified by other users, press F5 to refresh. </item>
    <item key="MSG0007">Can not be empty, please enter {0}. </item>
    <item key="MSG0008">{0} already exist. </item>
    <item key="MSG0009">New record created. </item>
    <item key="MSG0010">Change saved. </item>
    <item key="MSG0011">Data saved. </item>
    <item key="MSG0012">Batch data saved. </item>
    <item key="MSG0013">Data deleted. </item>
    <item key="MSG0014">Batch data deleted. </item>
    <item key="MSG0015">Do you want to delete?</item>
    <item key="MSG0016">Do you want to delete {0}?</item>
    <item key="MSG0017">Can not delete this record. </item>
    <item key="MSG0018">{0} can not be deleted. </item>
```

图 24-02 语言包里的内容参考

这是语言包里的内容参考，每个页面上的元素的多语言定义，中间用了“_”区分了每个窗体里的控件的名称。




```
en-US.xml

<!-- FrmLogin Begin  -->
<item key=“FrmLogin”>Login</item>
<item key=“FrmLogin_grpLogin”>Login</item>
<item key=“FrmLogin_lblUser”>&amp;User Name:</item>
<item key=“FrmLogin_lblPassword”>&amp;Password:</item>
<item key=“FrmLogin_btnRequestAnAccount”>&amp;Account</item>
<item key=“FrmLogin_btnConfirm”>&amp;Login</item>
<item key=“FrmLogin_btnExit”>&amp;Exit</item>
<item key=“FrmLogin_btnCancel”>&amp;Cancel</item>
<!-- FrmLogin End-->

<!-- FormModuleAdd Begin  -->
```

图 24-03 页面上的元素

文字内容需要设计时注意，中文的长度与英文的长度不一样，对齐方式等都需要仔细调整好，否则英文版就无法用了，乱了。



图 24-04 登录窗体中文版

页面上需要写一行代码，是否需要多语言功能的一行代码。




```
FrmLogin.cs   FrmLogin.cs [设计]
DotNet.WinForm.Login.FrmLogin
160                 this.txtPassword.Focus();
161             }
162             // 多语言国际化加载
163             this.Localization(this);
164             // 设置帮助
165             this.SetHelp();
166             // 设置按钮状态
167             this.SetControlState();
168             this.FormLoaded = true;
169             // 设置鼠标默认状态
170             this.Cursor = Cursors.Default;
171         }
172         #endregion
173 
174         private void FrmLogin_Load(object sender, EventArgs e)
175         {
176             if (!this.DesignMode)
177             {
178                 // 设置鼠标繁忙状态
179                 this.Cursor = Cursors.WaitCursor;
180                 try
181                 {
182                     // 加载窗体
183                     this.FormOnLoad();
184                 }
185                 catch (Exception ex)
186                 {
187                     this.ProcessException(ex);
188                 }
189                 finally
190                 {
191                     // 设置鼠标默认状态
192                     this.Cursor = Cursors.Default;
193                 }
194             }
195         }
```

图 24-05 是否需要多语言功能

在启动文件里、需要读取资源文件，多语言包资源。

```
Program.cs
DotNet.WinForm.Program                     Main()
73              }
74 
75              // 多语言国际化加载
76              ResourceManagerWrapper.Instance.LoadResources(Application.StartupPath + @"\Resource\Localization\");
77              // 从当前指定的语言包读取信息
78              AppMessage.GetLanguageResource();
79 
80              // 分级管理，按什么来计算，是否按编号计算孩子节点
81              // BasePermissionScopeManager.GetChildrensByCode = true;
```

图 24-06 读取资源文件

配置文件里，需要指定默认的语言，如下：

Config.xml

```
<!-- 客户等相关信息配置 -->
<add key="CustomerCompanyName" value="杭州吉日软件"/>
<add key="ConfigurationFrom" value="UserConfig"/>
<add key="SoftName" value="DotNet"/>
<add key="SoftFullName" value="走火入魔.NET通用快速开发架构"/>
<add key="Version" value="3.0" />
<add key="RootMenuID" value="Water"/>
<add key="CurrentLanguage" value="en-US"/>

<!-- 启用分级管理范围权限设置 -->
<add key="UseScopeAdmin" value="True"/>
<!-- 启用逐级授权 -->
<add key="UseLicensedPermission" value="false"/>
<!-- 登录窗体 -->
<add key="LoginAssembly" value="DotNet.WinForm.Login"/>
<add key="LoginForm" value="FrmLogin"/>
<add key="MainForm" value="SDIMainForm"/>
<!-- 默认加载用户 -->
<add key="LoadUser" value="true"/>
<!-- 允许用户注册 -->
<add key="AllowUserRegister" value="false"/>

<add key="RegisterKey" value="12508090402010371027201200790750
```

图 24-07 指定默认的语言

登录窗体中文版运行效果图如下：

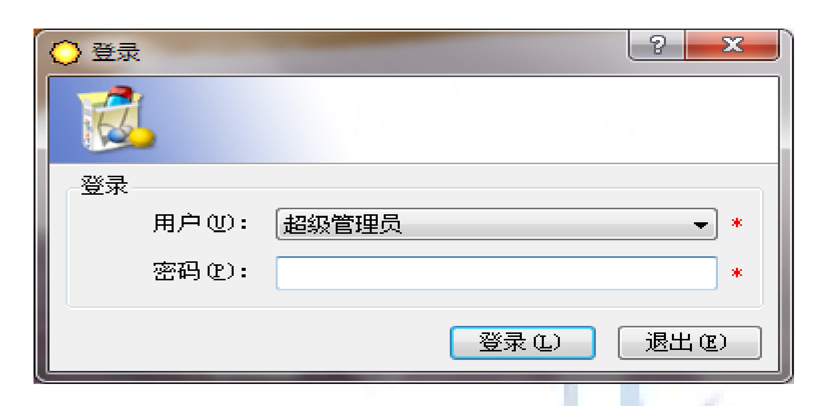


图 24-08 登录窗体中文版

登录窗体英文版运行效果图入下：

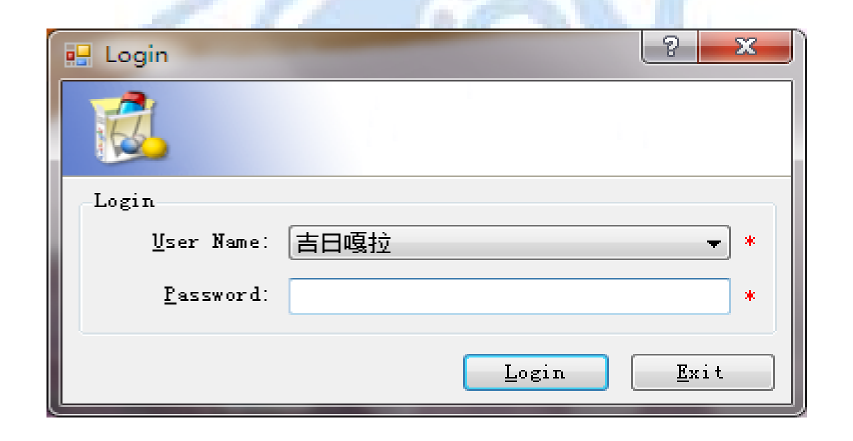


图 24-09 登录窗体英文版



第30章 代码生成器

有了数据库结构设计文档后，若直接按这个设计文档生成相应的代码，工作效率还是会显著提高，一方面可以从这个无聊的、没有什么创造性的工作里解放出来，另一方面生成好的代码质量也高一些，后期的维护也方便一些。

你想让 10 个人写出来的代码都像一个人一样还是想当的困难，若没有超强的管理能力甚至是可以说，不太可能看上去像一个人代码，各种风格都会有。

盖大楼先有图纸，再进行施工一样，做软件项目，还是先有设计后有代码比较好，修改设计上花费的代价远比修改代码上花费的成本会少很多。

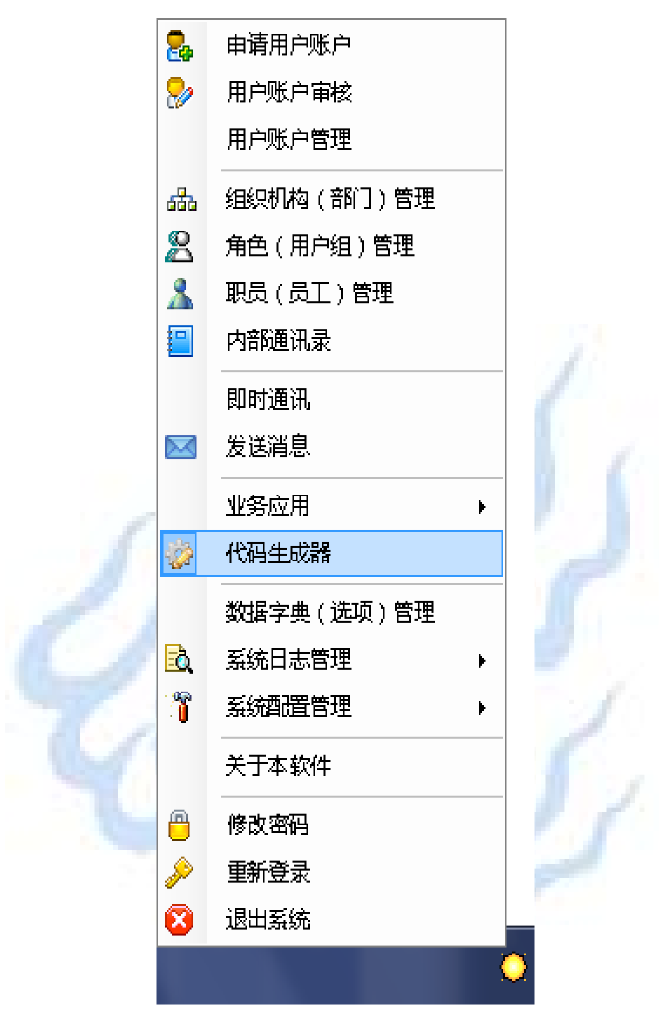


图 25-01 代码生成器菜单

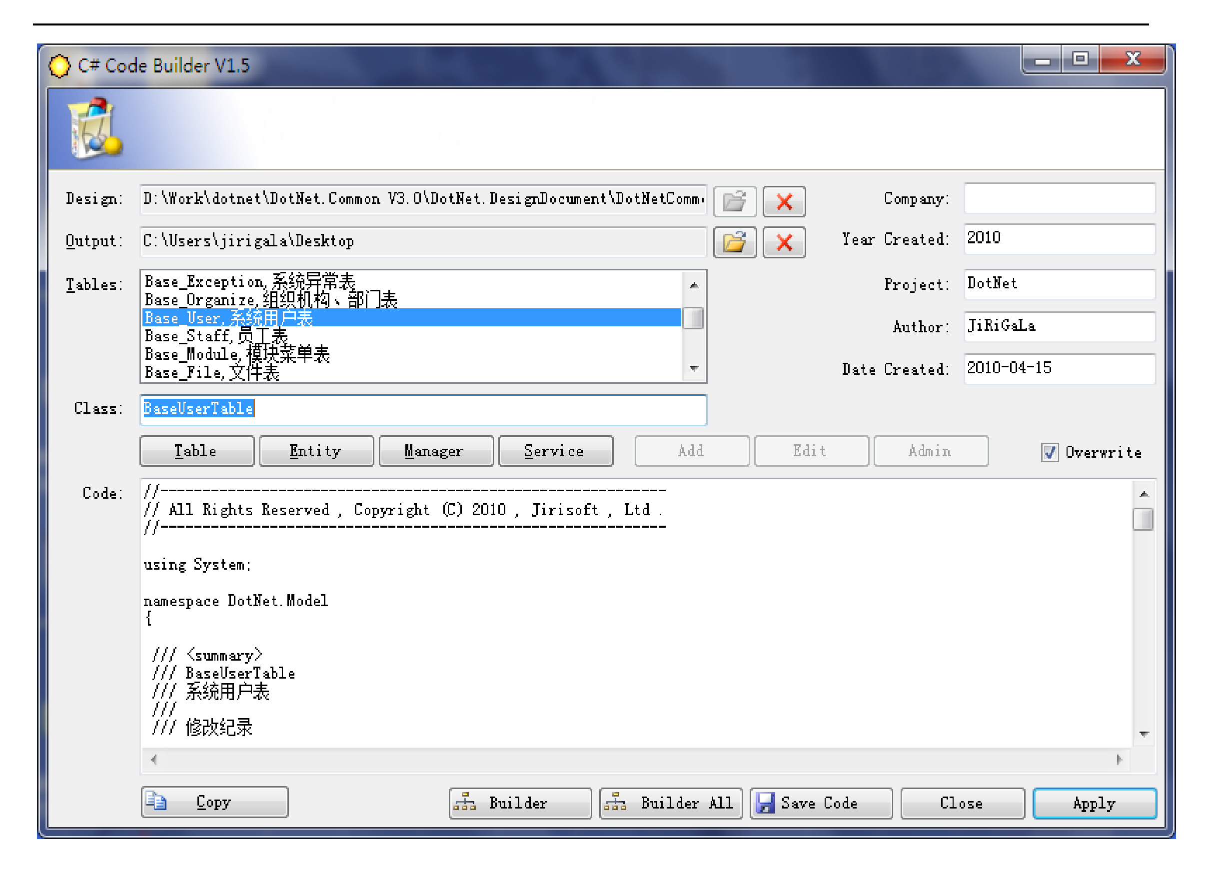


图 25-02 代码生成器窗体



第31章 典型业务案例应用

31.1 文档管理

文档管理模块可以灵活管理公司的内部文档，可以进行添加、修改、删除、移动文档的操作，可以灵活组织文档目录，界面如下图。

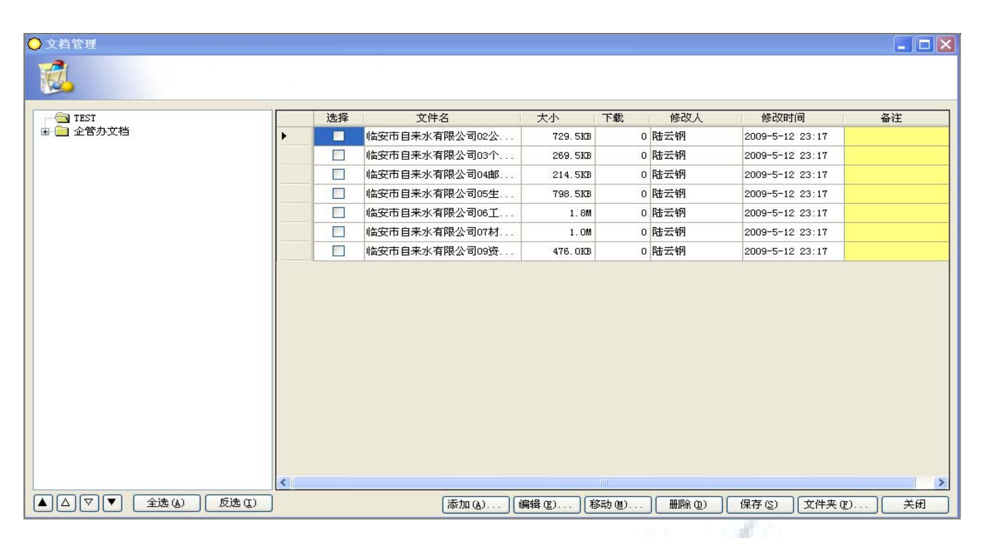


图 26-1-1 文档管理界面

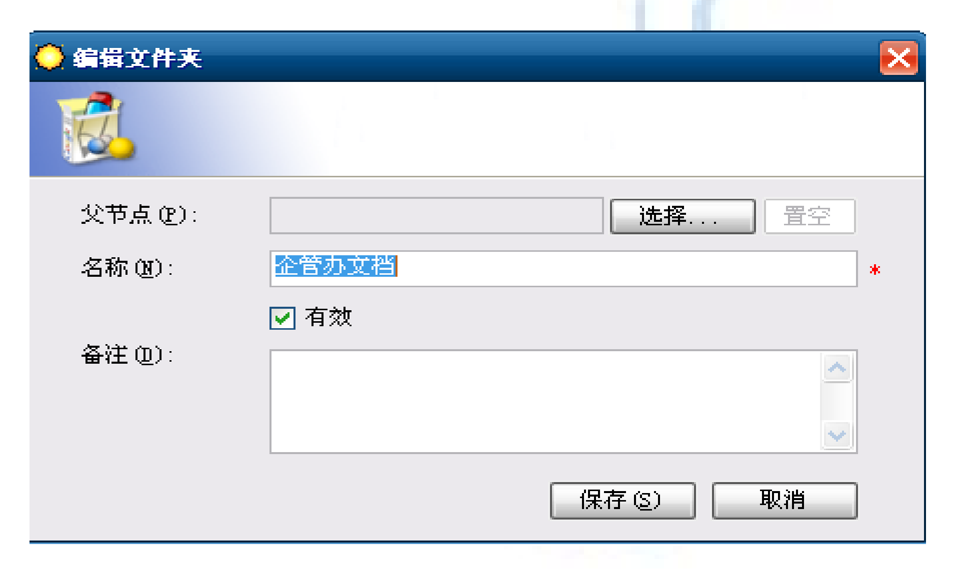


图 26-1-2 编辑文件夹

文件夹可以支持拖动功能，文件也支持拖动功能，文件夹、文件支持重命名，从技术的角度、本模块功能可以作为文件保存的数据库的技术实现的一整套的入门参考程序。

31.2 名片管理

名片管理模块可以管理整个公司的共用名片信息，可以进行添加、修改、删除操作、导出数据操作。主界面运行效果图如下。

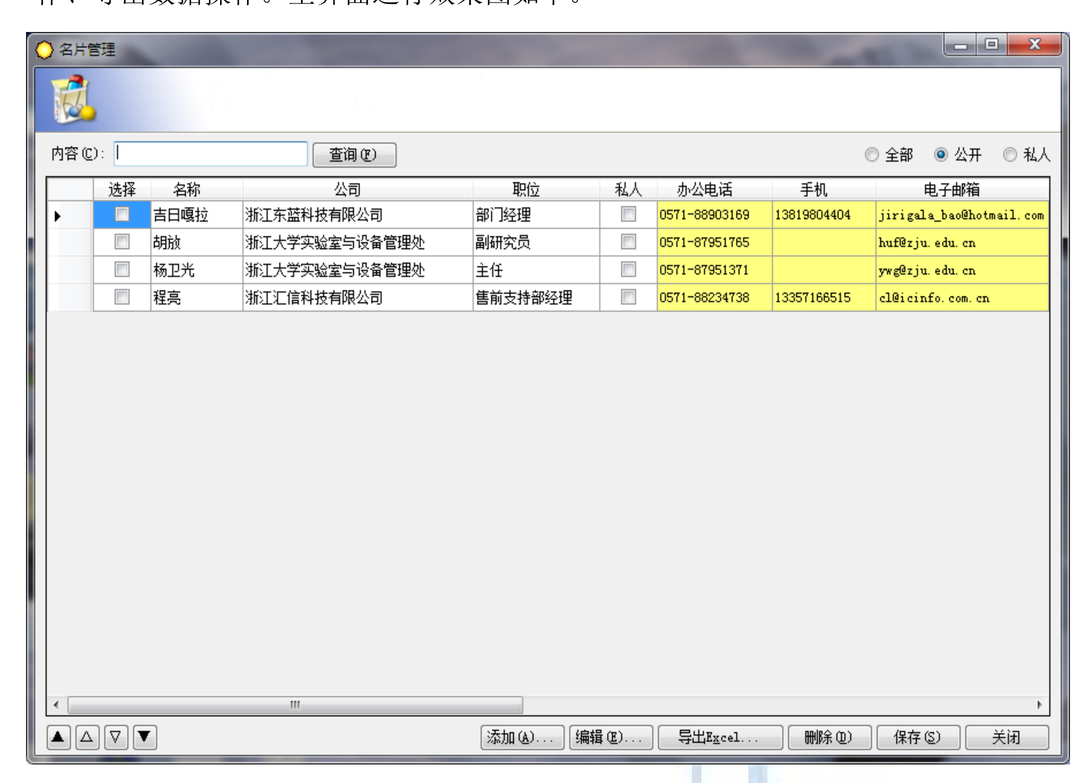


图 26-2-1 名片管理窗体



添加名片功能参考如下图。

图 26-2-2 添加名片窗体

修改名片信息效果如下图。



图 26-2-3 编辑名片窗体

名片管理程序还可以单独运行，可以作为一个最简单的 C/S 结构程序的开发入门参考，作为快速入门的经典教程。

第32章 开发登录功能过程分析

32.1 系统登录功能的分工

32.1.1 市场营销人员

32.1.2 决策老板

32.1.3 项目经理

32.1.4 系统架构师

32.1.5 美工设计人员

32.1.6 程序员

32.1.7 质量保障员

32.1.8 测试人员

32.1.9 文档管理员

32.1.10 客户经理

32.1.11 实施工程师

做一个登录系统需要注意的事项有以下几点：

1) 这个程序是否已经有人做好了？还是有标准的模块？还是有曾经留下的文档？是否是崭新的开始？还是修改别人的程序？
2) 要为什么系统做登录窗体？都需要注意什么？谁能把需求讲清楚？做到什么样子才能满足要求？
3) 密码是否允许为空？应该系统里可以配置，允许空密码的做法虽在开发阶段，调试程序很便捷，但系统上线后，空密码会引起重大安全隐患。

4) 登录用户名是否可以修改？若登录名不允许修改，系统也会有些不友善，不太人性化，天天用一个自己不喜欢的、不习惯的用户名，换成你会是什么滋味呢。
5) 登录次数限制？防止别人恶意输入很多次密码，暴力破解密码。
6) 登录时的安全加强，防止网络灌水机、垃圾广告等，例如登录时都需要有图片验证码等，大多是为了防止垃圾广告注入程序的。
7) 用户长度限制？若本来系统就做得不够安全，还能输入很长的用户名，很容易被黑客利用这个漏洞。
8) 密码长度限制？系统应该可以设置最短的系统密码长度，当然能限制密码强度要求也是更好的、更理想的。
9) 用户名、密码的大小写处理相关问题。
10) 后台用户密码的存储问题，明码保存还是加密保存？如何防止数据库管理员窃取密码行为？
11) 控制按钮的状态？例如最大化、最小化按钮在登录窗体是没多大必要的，本来就这么点儿的窗体，还要那么多功能干啥呢？
12) 登录 IP 地址限制？例如一些重要的信息系统，例如财务系统，不是随便一台电脑都可以登录，人事薪资系统、客户服务系统也是，不应该哪个电脑都能登录系统，这些都是公司的高度机密信息。
13) 登录系统是一个很重大的事件、就像是公安的作案时间一样，若一个系统出了什么问题，都不能知道作案时间，作案地点，那是比较麻烦了，所以谁什么时候，在什么地点登录都需要有日志记录，那些搞破坏的、黑客攻击的，更应该都能记录日志才可以。
14) 一个帐号只能登录一次？有些由于特定的业务上的要求，不允许一个帐号同时登录系统，这时候系统应该能有设置，是否允许用户多次登录的选项。
15) 登录程序的输入检查？就是例如密码不允许空时不输入密码是不能进行登录，用户名不输入、不选择时当然不能登录，这些都应该在页面输入检查有效性时就检查好。
16) 光标停留位置的处理，例如登录时用户名没输入应该光标停留在用户名上、若已经用户名有了，那应该停留在密码上，若用户名、密码都有了，光标应该停留在登录上。

17) 回车按钮的友善处理，中国人大多都习惯了按回车按钮，虽然键盘上有 Tab 键是控制光标用的，但是全国人民都习惯了按回车了，所以还是满足一下大家的实际需要比较好。
18) 退出键盘的优化，按 ESC 应该能退出系统才比较好。
19) 页面的布局是否、控件的摆放位置是否正确？控件之间的间距是否合理？按钮的大小是否合理？
20) 客户的实际生产环境往往与开发环境有很大的不同，是否考虑了鼠标没装或者鼠标失灵时，系统是否可以好用？有没考虑好快捷键操作。
21) 用户名密码是否可以保存？若用的是自己的电脑、每人一台电脑，那就没必要每次都输入用户名、密码，所有能保存用户名密码的功能比较好。
22) 用户名、密码保存时是否要进行加密功能？若万一电脑被别人用了，老板的密码被别人轻易的知道了，是不是有些不太好？是否适当的进行用户密码的安全性。
23) 用户名、密码保存在哪里？注册表里？文本文件里？XML 文件里？业务系统是做什么的？应对用户的实际情况下应该怎么做会更合理？更贴切？
24) 登录程序有没有一定的兼容性？例如 C/S 架构的与 B/S 架构的页面是否都可以调用同一个登录服务程序？总不能每个都写一套吧？
25) 登录程序是否支持多语言支持？例如是一个跨国集团，又有中国人、日本人、美国人、西班牙人、法国人，或者用户群体是国际性的，这时候也需要支持多语言功能的需求。
26) 登录程序是否可以用其他的信息系统的用户名密码、修改身份验证方式后，会不会对系统的影响是最低的？
27) 单点登录？若公司里有很多套管理系统，最好用户名密码能保持一致比较好，可以用同一套用户名密码可以登录各个信息系统比较友善一些，就向每个人的身份证一样，没必要到每个地方都换一下身份证，到每个公司都要换身份证吧。
28) 登录时间限制？有些要求严格的系统，只能在上班时间登录，银行业务系统会不会是有这样的限制、倒不是很清楚。
29) 每个月必须换密码？有些要求严格的财务系统，需要每个月更换密码的安全要求等等。

30) 在公司登录系统？在家里登录系统？是否可以满足移动办公的要求，在公司登录时很可能是登录到局域网里，在家里登录时很可能登录到广域网的服务器上，系统是否可以灵活进行设置？
31) 登录成功后，应该调用哪个窗体？是否还要调用哪个方法？还需要注意什么？
32) 登录成功后，是否需要有提示信息？应该怎么提示客户？
33) 登录失败后的提示信息？
34) 程序运行过程中发生的异常怎么处理？客户端发生的呢？服务器端发生的呢？
35) 是否可以在系统的升级、优化、变更时对登录程序的影响设计到最小？考虑、设计、实现得周全？
36) 登录程序的数据传输是否存在安全效率？会不会给小黑客截取？存在安全隐患？客户是否有这么严的要求？
37) 登录程序与服务器的交互次数是否最优？传输的数据量是否最小？
38) 登录成功后都要处理那些事情？调用那些处理函数？登录失败后，又要进行那些处理？
39) 相关的成果物（文档、代码、数据库）存放哪里？自己的电脑里？公司的服务器上？同时多个人开发如何解决版本问题，程序修改错了怎么办？误删除了怎么办？
40) 登录程序的可替换？不同的业务系统，不同的客户的个性化需求都比较多一些，有时候又不得不去满足客户的要求，不管客户的需求是对的还是错的，这时候要求登录程序是可替换的、用户到底想用那个登录程序，可以进行配置。

若日常程序开发工作中，用这个细腻的思想去做其他业务系统，照样可以做出质量过关的程序，当然对客户的业务需求是否有深入的了解，你的设计是否贴近客户的实际需要，那就看你的理解能力、业务知识积累、用心用功程度上的差别了。

32.2 注意事项建议

32.2.1 分工明确原则

32.2.2 沟通效率问题

32.2.3 决策效率问题

32.2.4 工作量问题

32.2.5 产品质量问题

32.2.6 人员变动问题

32.2.7 绩效考核问题

32.2.8 工作成果问题

32.3 通过实现登录窗体的反思

为什么软件公司一天到晚都忙个不停？就是因为这么一个简单的登录程序，都需要考虑几十个因素，更何况功能复杂的业务系统，不大可能就这么 2 个输入项目，2 个按钮这么简单的还是很少。

为什么客户想要个简单的系统，就是开发不好？因为就要个这么简单的登录功能彻底做得好都不容易，更何况其他比这个相对复杂的业务系统了，而且需求都没深入了解、也没把工作难点、重点都没分析清楚的前提下，怎么可能保证把业务系统做好？

为什么项目型软件公司很累?赚钱很难？因为一个系统的登录做好都要做这么工作，那开发个 10 个 80 个系统，那么每个功能都想做好，那需要付出很多代价，需要更强的管理能力、项目控制能力，需要有更敬业的员工在默默无闻的奉献。

为什么很多软件公司，做好软件项目很难？能深入研究、静心写程序，不断维护软件系统的人，而且还喜欢无聊程序的人还是少一些，过于枯燥，所以很多人都会选择放弃，甚至是当老板的也会放弃。



软件公司为什么需要那么多的程序员？大规模的软件为什么需要很多人去做？因为一个很简单的登录窗体都需要很多很多细节功能处理的精密才能是最后经得起考验，用户也喜欢用，也能符合用户的实际需要，所以需要很多很多人去精心维护好每个功能模块，甚至是需要多年的精心维护不断改进后，才能更加适应客户的实际需要。

为什么会有几千几万人的 IT 公司，他们都在忙乎什么？他们可能大多都在忙着处理类似的 N 个业务系统中的 N*N 个细节问题上，深入分析这些任务、分配这些任务、测试确认验收这些任务、追踪督促这些任务的完成情况等等。

软件的质量与其他行业一样，细节决定成败？别墅也是房子、茅草屋也是房子，都可以睡觉过夜，但是太多的细节上的差别了，豪华车也是车子，经济型低档车也是车，他们的区别也在于细节，只有精心打造每个环节，才能保证商品的整体的质量是优良的，拿一个车子打个比喻，方向盘、刹车也偶尔会故障，那会是什么情况？软件产品其实也是一样的道理。

管理类软件公司需要什么样的软件人才？什么人才更容易适应软件公司？说白了你能把一个登录程序分析得深入，而且还愿意分析客户的业务知识，那你应该是软件公司欢迎的人才，软件公司更喜欢思路严谨、缜密、有条有理的，更喜欢有一定的管理能力，有成本意识、有时间管理意识的人才。



第33章 客户经常需要的开发技能

33.1 打印功能

33.2 编号产生功能

33.3 查询功能

33.4 大数据量快速分页功能

33.5 能快速便捷输入数据功能

33.6 模块（菜单）可以灵活配置

33.7 数据导入导出功能

33.8 图标统计

33.9 第三方接口程序

33.10 程序需要经得起折腾

33.11 导入原有系统的数据

33.12 数据导入导出功能

33.13 漂亮的界面

第34章 产品报价、技术支持、售后服务

34.1 产品及服务报价

淘宝网店地址 http://jirigala.taobao.com/ 上购买相应的功能软件、服务后。

一）产品报价部分：

免费提供部分：网上发布的相关技术文章、例子代码、视频教程，案例分析等。

体验版（498 元）：提供走火入魔通用权限管理系统数据库文件 + C\S 权限配置程序，不提供源码，操作说明文档一份+相关项目源代码视频一份，仅提供 QQ 群技术支持。可以在自己开发的软件里集成权限组件，以体验产品功能。用户不得恶意非法反编译所获取的相关 DLL 文件,更替相关版权信息，与走火入魔配套的相关数据库及操作系统相关软件，请购买者自行解决，因运行环境使用不当，造成软件运行故障等相关问题，本司一律不负相关责任。但是可以提供咨询，收取一定的费用，帮助解决相关问题。用户获取所有的 Dll 文件均受国家知识产权法保护，用户不得在未授权的情况下，用作商业用途，违者必究。若需要用作商业用途，请联系杭州海日涵科技有限公司相关人员，签订相关商业之合同。

学习版（998 元）：提供走火入魔通用权限管理系统数据库文件 + C\S 权限配置程序源码，操作说明文档一份+相关项目源代码视频一份，仅提供 QQ 群及 QQ 技术支持，可以在自己开发的软件里集成权限组件及相关文档，本版本是针对软件业务爱好者，以体验及学习系统功能与及设计思想和结构为主。与走火入魔配套的相关数据库及操作系统相关软件，请购买者自行解决，因运行环境软件使用不当，造成软件运行故障等相关问题，本司一律不负相关责任。但是可以提供咨询，收取一定的费用，帮助解决相关问题。用户不得将源码用于商业行为，更不得更改相关版权信息，违者必究。若需要用作商业用途，请联系杭州海日涵科技有限公司相关人员，签订相关商业之合同。

开发版（4998 元）：提供走火入魔通用权限管理系统数据库文件 + C\S 权限配置程序源码+ B\S 接口程序源码 + 数据库设计文档+操作说明文档一份+相关项目源代码视频一份，可以在自己开发的软件里集成权限组件及相关文档，不能用于公司级商业行为，不得单独销售权限系统， 本版本是针对软件个人开发者为主， 提供安装、实施服务，QQ 群，QQ， Email 技术支持，只限购买者本人提供技术支持服务，与走火入魔配套的相关数据库及操作系统软件，请购买者自行解决，因运行环境使用不当，造成软件运行故障等相关问题，本司一律不负相关责任。但可以提供相关问题的咨询服务，经双方确认，收取一定的费用，该版本付费后，签订纸质的"走火入魔通用权限管理系统（开发版）销售合同书",按合同约定好的双方必须遵守的各自权责相关事宜。受法律保护。

。

专业版（29998 元）：提供走火入魔通用权限管理系统全部源码+数据库文件+ B\S 接口程序源码 + 数据库设计文档+操作说明文档一份+相关项目源代码视频一份+ B\S 管理系统参考用例，一年内的技术支持 + 二年的升级服务，可以在自己开发的软件里集成权限组件及相关文档，保留其版权相关信息，但是不得单独销售权限系统，该版本是针对软件公司、技术咨询公司为主， 提供安装、实施服务， 全天 QQ 在线技术支持或 Email 技术支持，提供全天的电话技术支持、可享受权限技术相关的疑难问题解答服务。与走火入魔配套的相关数据库及操作系统软件，请购买者自行解决，因运行环境使用不当，造成软件运行故障等相关问题，本司一律不负相关责任。但可以免费提供相关问题的咨询服务。此版本付费时，签订纸质的"走火入魔通用权限管理系统（专业版）销售合同书",按合同约定好的双方必须遵守的各自权责相关事宜。受法律保护。

专家版（59998 元）：提供走火入魔通用权限管理系统全部源码+数据库文件 + B\S 接口程序源码 + 数据库设计文档+操作说明文档一份+相关项目源代码视频一份 + B\S 管理系统参考用例，三年内的技术支持 + 十年的升级服务，可以在自己开发的软件里集成权限组件及相关文档，保留其版权相关信息，但是不得单独销售权限系统，与走火入魔配套的相关数据库及操作系统软件，请购买者自行解决，因运行环境使用不当，造成软件运行故障等相关问题，本司一律不负相关责任。但可以免费提供相关问题的咨询服务，此版本是针对最终客

户、政府单位、大型企业、高校等为主， 提供现场安装、实施服务、培训服务，全天 QQ 在线技术支持或 Email 技术支持，提供全天的电话技术支持、可享受权限技术相关的疑难问题解答服务，并提供现场的深入的系统分析讲解，完整的技术方案讲解等服务，，此版本付费时，签订纸质的“走火入魔通用权限管理系统（专家版）销售合同书”,按合同约定好的双方必须遵守的各自权责相关事宜。受法律保护。

二）技术服务报价部分：

项目急救 3000 元/天(工作 8 个小时内)

项目管理 2000 元/天(工作 8 个小时内)

技术培训 2000 元/天(工作 8 个小时内)

项目验收 2000 元/天(工作 8 个小时内)

项目测试 1500 元/天(工作 8 个小时内)

代码检查 1000 元/天(工作 8 个小时内)

开发平台、开发框架讲解(技术指导) >= 2000 元/天*5 天以上项目

数据库设计 2000 元/天(工作 8 个小时内)

34.2 联系方式、售后服务、错误反馈

一）**QQ 技术支持**：

2520 - 56973

二）**MSN 技术支持**：

JiRiGaLa_Bao#Hotmail.com

三）**Email 技术支持**：

JiRiGaLa_Bao#Hotmail.com

四）电话技术支持：

138 – 5816 – 3011

五）联系人：

吉日嘎拉（吉日先生）